

TUGAS AKHIR - KI141502
**RANCANG BANGUN PEMBANGKIT SKENARIO
DINAMIS PADA PERMAINAN *BANGJO PUZZLE*
BERBASIS ANDROID**

BULAN RAHMAT SETYADI
NRP 5112100150

Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.
Wijayanti Nurul Khotimah, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

(Halaman Ini Sengaja Dikosongkan)



TUGAS AKHIR - KI141502

**RANCANG BANGUN PEMBANGKIT SKENARIO
DINAMIS PADA PERMAINAN *BANGJO PUZZLE*
BERBASIS ANDROID**

**BULAN RAHMAT SETYADI
NRP 5112100150**

**Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.
Wijayanti Nurul Khotimah, S.Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**

(Halaman Ini Sengaja Dikosongkan)



FINAL PROJECT- KI141502

DESIGN AND DEVELOPMENT OF DYNAMIC SCENARIOS ON BANGJO PUZZLE GAME ON ANDROID

**BULAN RAHMAT SETYADI
NRP 5112100150**

**Advisor
Imam Kuswardayan, S.Kom., M.T.
Wijayanti Nurul Khotimah, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2018**

(Halaman Ini Sengaja Dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN PEMBANGKIT SKENARIO DINAMIS PADA PERMAINAN *BANGJO PUZZLE* BERBASIS ANDROID

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

BULAN RAHMAT SETYADI
NRP. 5112 100 150

Disetujui oleh Dosen Pembimbing Tugas Akhir
Imam Kuswardayan, S.Kom., M.T.
NIP: 19761215 200312 1 001 (pembimbing 1)

Wijayanti Nurul Khotimah, S.Kom., M.Sc.
NIP: 19860312 201212 2 004 (pembimbing 2)

SURABAYA
JANUARI, 2018

(Halaman Ini Sengaja Dikosongkan)

RANCANG BANGUN PEMBANGKIT SKENARIO DINAMIS PADA PERMAINAN BANGJO PUZZLE BERBASIS ANDROID

Nama Mahasiswa : Bulan Rahmat Setyadi
NRP : 5112100150
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing I : Imam Kuswardayan, S.Kom., M.T.
Dosen Pembimbing II: Wijayanti Nurul Khotimah, S.Kom., M.Sc.

ABSTRAK

Permainan adalah salah satu media hiburan yang cukup populer. Sifatnya yang praktis dan mudah digunakan menjadikan permainan banyak diminati oleh berbagai kalangan. Terdapat berbagai macam jenis permainan yang beredar di masyarakat. Salah satunya adalah permainan puzzle. Namun, seringkali skenario yang ada pada permainan puzzle bersifat statis. Skenario yang statis akan menyebabkan pengguna merasa cepat bosan karena telah menjumpai skenario yang sama berkali-kali.

Tugas akhir ini merancang dan membangun sebuah permainan puzzle dengan skenario yang dinamis. Pengguna diharuskan menyelesaikan skenario-skenario yang tersaji dalam tiga tingkat, yaitu easy, medium, dan hard. Metode yang digunakan dalam pengacakan skenario adalah Midpoint Displacement. Setelah itu, dilakukan penentuan jalur dengan menggunakan algoritma Brute Force. Pengujian yang dilakukan adalah pengujian pembangkitan puzzle dan pengujian penyelesaian skenario. Hasil pengujian menunjukkan bahwa pembangkitan puzzle dan penyelesaian skenario telah berhasil dilakukan.

Kata kunci: Algoritma Brute Force, Permainan Puzzle, Midpoint Displacement

(Halaman Ini Sengaja Dikosongkan)

DESIGN AND DEVELOPMENT OF DYNAMIC SCENARIOS ON BANGJO PUZZLE GAME ON ANDROID

Student Name : Bulan Rahmat Setyadi
NRP : 5112100150
Major : Departement of Informatics FTIK-ITS
Advisor I : Imam Kuswardayan, S.Kom., M.T.
Advisor II : Wijayanti Nurul Khotimah, S.Kom., M.Sc.

ABSTRACT

Game is one of the entertainment media that is quite popular. Many people choose game as because it's practical and easy to use. There are various types of games that has been published. One of them is puzzle game. The problem is, often the scenario in puzzle game is static. A static scenario will cause users to feel bored quickly because they have encountered the same scenario many times.

This final project is design and development a puzzle game with dynamic scenarios. Users are required to complete the scenarios that represented in three levels, easy, medium, and hard. The method used in randomizing scenarios is Midpoint Displacement. After that, Brute Force algorithm will be used for determining the solution path. Testing of puzzle generation and testing of scenario completion will be conducted. The result show that puzzle generation and scenario completion have been applied successfully.

Keywords: Brute Force Algorithm, Puzzle Game, Midpoint Displacement

(Halaman Ini Sengaja Dikosongkan)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang telah memberikan rahmat dan petunjuk-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul:

RANCANG BANGUN PEMBANGKIT SKENARIO DINAMIS PADA PERMAINAN *BANGJO PUZZLE* BERBASIS ANDROID

Pengerjaan tugas akhir ini merupakan sarana belajar bagi penulis untuk dalam bidang ilmu informatika, khususnya dalam rumpun mata kuliah Interaksi Grafik dan Seni yang diwujudkan dalam bentuk permainan.

Penulis menyadari bahwa tugas akhir ini dapat terselesaikan disebabkan oleh bantuan berbagai pihak. Tidak ada hal lain yang dapat penulis berikan selain mengucapkan terima kasih kepada semua yang telah terlibat dalam penyusunan tugas akhir ini. Penulis mengucapkan terima kasih kepada:

1. Allah SWT, atas kehendak-Nya tugas akhir ini dapat terselesaikan dengan baik.
2. Alm. Agus Wahyudi dan dr. Setyawati, selaku orang tua penulis yang senantiasa mendukung setiap langkah penulis.
3. Agung Teguh Setyadi, selaku saudara penulis yang senantiasa gembira apabila penulis mengalami kesusahan dalam menyusun tugas akhir.
4. Dimas Putri Supawesti yang pada kenyataannya tidak membantu apa-apa.
5. Bapak Dr. Darlis Herumurti, S.Kom., M.Kom., selaku Ketua Departemen Informatika ITS.
6. Bapak Radityo Anggoro, S.Kom., M.Eng.Sc., Ph.D., selaku koordinator tugas akhir di Departemen Informatika ITS.

7. Bapak Imam Kuswardayan, S.Kom., M.T., selaku pembimbing I tugas akhir yang telah memberikan banyak waktu dan ilmu yang bermanfaat kepada penulis, semoga ilmu tersebut senantiasa diganjar pahala yang jariyah.
8. Ibu Wijayanti Nurul Khotimah, S.Kom., M.Sc., selaku pembimbing II tugas akhir yang telah memberikan banyak waktu dan ilmu yang bermanfaat kepada penulis, semoga ilmu tersebut senantiasa diganjar pahala yang jariyah.
9. Bapak dan ibu dosen di Departemen Informatika ITS yang telah memberikan ilmunya kepada semua mahasiswanya, semoga ilmu tersebut senantiasa diganjar pahala yang jariyah.
10. Seluruh staf dan karyawan Departemen Informatika ITS yang telah memberikan bantuan kepada penulis.
11. Rekan-rekan angkatan 2012 penulis, baik yang telah menyelesaikan studi maupun yang belum.
12. Rekan-rekan kloter akhir 2012, Dzul, Soca, Fatih, Ivan, Yoan, dan Naufal terima kasih telah menjadi teman penulis di akhir waktu penulis mengemban ilmu di Departemen Informatika ITS.
13. Akbar Anugerah Syahdamar, S.Kom., sahabat penulis yang senantiasa membantu penulis dari jurang kesulitan dalam mengemban ilmu. Sukses untukmu dunia dan akhirat.
14. Yusuf Aziz Henny Tri Yudhantoro, S.Kom., sahabat penulis dari SD, SMP, SMA, dan sekarang sekuliahan. Sukses untukmu dunia dan akhirat
15. Rekan-rekan di GSM Production, Burhan, Reva, Bagus, Dicho, dan Yusron terima kasih telah mengajari saya bahwa kita tidak boleh tergantung pada teman, karena pada akhirnya kehidupan kita ditentukan oleh langkah kita sendiri.

Penulis memohon maaf apabila terdapat kekurangan dalam penulisan tugas akhir ini. Kritik dan saran selalu penulis tunggu untuk pembelajaran selanjutnya. Semoga tugas akhir ini dapat bermanfaat di dunia maupun di akhirat. Aamiin.

Surabaya, Januari 2018

Bulan Rahmat Setyadi

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
1.6 Metodologi.....	3
1.7 Sistematika Penulisan Laporan Tugas Akhir	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 <i>Game Maker</i>	7
2.2 <i>Midpoint Displacement</i>	8
2.3 <i>Algoritma Brute Force</i>	11
BAB III ANALISIS DAN PERANCANGAN	13
3.1 Analisis Sistem	13
3.1.1 Analisis Kebutuhan Fungsional	13
3.1.2 Deskripsi Umum Perangkat Lunak	14
3.2 Perancangan Sistem	14
3.2.1 Perancangan Fungsionalitas Aplikasi	14
3.2.2 Perancangan Skenario Kasus Penggunaan.....	14
3.2.3 Perancangan Antarmuka Pengguna	21
3.2.4 Perancangan Kontrol Permainan	24
3.2.5 Perancangan Aturan Permainan.....	24
3.2.6 Perancangan Metode	24
BAB IV IMPLEMENTASI	41
4.1 Lingkungan Implementasi.....	41

4.2	Implementasi Permainan.....	41
4.2.1	Implementasi Tampilan Menu Utama	41
4.2.2	Implementasi Tampilan Pemilihan Tingkat.....	42
4.2.3	Implementasi Tampilan Bermain Permainan	43
4.2.4	Implementasi Tampilan <i>Instruction</i>	43
4.2.5	Implementasi Tampilan <i>Credit</i>	44
4.2.6	Implementasi Kasus Penggunaan Memilih Tingkat Permainan	45
4.2.7	Implementasi Kasus Penggunaan Mengulangi Tingkat Skenario	45
4.2.8	Implementasi Kasus Penggunaan Melihat Instruksi Permainan	46
4.2.9	Implementasi Kasus Penggunaan Melihat <i>Credit</i> Permainan	46
4.2.10	Implementasi <i>Midpoint Displacement</i>	47
4.2.11	Implementasi Penentuan Jalur.....	48
BAB V	PENGUJIAN DAN EVALUASI.....	51
5.1	Lingkungan Uji Coba	51
5.2	Pengujian Pembangkitan <i>Puzzle</i>	51
5.2.1	Skenario Uji Pembangkitan <i>Puzzle</i>	51
5.2.2	Hasil Pengujian Pembangkitan <i>Puzzle</i>	52
5.3	Pengujian Aturan Permainan <i>Bangjo Puzzle</i>	53
5.3.1	Skenario Uji Aturan Permainan <i>Bangjo Puzzle</i>	53
5.3.2	Hasil Pengujian Aturan Permainan <i>Bangjo Puzzle</i>	53
5.4	Pengujian Penyelesaian Skenario.....	56
5.4.1	Skenario Uji Penyelesaian Skenario.....	56
5.4.2	Hasil Uji Penyelesaian Skenario	56
5.5	Pengujian Fungsionalitas Permainan	57
5.5.1	Skenario Uji Fungsionalitas Permainan	57
5.5.2	Pengujian Fungsionalitas Kasus Penggunaan.....	58
5.5.3	Hasil Pengujian Fungsionalitas	62
5.6	Pengujian Pengguna	62
5.6.1	Skenario Uji Pengguna.....	62
5.6.2	Daftar Penguji Perangkat Lunak	63
5.6.3	Hasil Penilaian Pengguna.....	63

5.6.4	Kesimpulan Hasil Pengujian Pengguna	66
BAB VI	KESIMPULAN DAN SARAN.....	67
6.1.	Kesimpulan	67
6.2.	Saran	68
DAFTAR PUSTAKA.....		69
LAMPIRAN 1		71
LAMPIRAN 2		77
LAMPIRAN 3		81
BIODATA PENULIS		87

(Halaman Ini Sengaja Dikosongkan)

DAFTAR GAMBAR

Gambar 3.1 Diagram Fungsionalitas Permainan	14
Gambar 3.2 Diagram Aktivitas Memilih Tingkat Permainan.....	18
Gambar 3.3 Diagram Aktivitas Bermain Permainan.....	19
Gambar 3.4 Diagram Aktivitas Mengulangi Tingkat Skenario	19
Gambar 3.5 Diagram Aktivitas Melihat Instruksi Permainan	20
Gambar 3.6 Diagram Aktivitas Melihat <i>Credit</i> Permainan	20
Gambar 3.7 Perancangan Tampilan Menu Utama	21
Gambar 3.8 Perancangan Tampilan Pemilihan Tingkat	22
Gambar 3.9 Perancangan Tampilan Bermain Permainan	22
Gambar 3.10 Perancangan Tampilan <i>Instruction</i>	23
Gambar 3.11 Perancangan Tampilan <i>Credit</i>	23
Gambar 3.12 Daftar Objek Permainan <i>Bangjo Puzzle</i>	25
Gambar 3.13 <i>Midpoint Displacement</i> (Langkah 1).....	26
Gambar 3.14 <i>Midpoint Displacement</i> (Langkah 2).....	26
Gambar 3.15 <i>Midpoint Displacement</i> (Langkah 3).....	27
Gambar 3.16 <i>Midpoint Displacement</i> (Langkah 4).....	27
Gambar 3.17 <i>Midpoint Displacement</i> (Langkah 5).....	28
Gambar 3.18 Hasil Proses <i>Midpoint Displacement</i>	28
Gambar 3.19 Hasil Pengubahan Angka Menjadi Nol dan Satu	30
Gambar 3.20 Penentuan Posisi <i>Start</i>	31
Gambar 3.21 Proses Penentuan Jalur Penyelesaian Melalui Titik <i>D1</i>	32
Gambar 3.22 Jalur Penyelesaian Pertama Pada Titik <i>D1</i>	32
Gambar 3.23 Jalur Penyelesaian Kedua Pada Titik <i>D1</i>	33
Gambar 3.24 Jalur Penyelesaian Ketiga Pada Titik <i>D1</i>	33
Gambar 3.25 Jalur Penyelesaian Keempat Pada Titik <i>D1</i>	34
Gambar 3.26 Jalur Penyelesaian Kelima Pada Titik <i>D1</i>	34
Gambar 3.27 Hasil Penentuan Jalur Penyelesaian Pada Titik <i>D1</i> ...	35
Gambar 3.28 Hasil Penentuan Jalur Penyelesaian Pada Titik <i>E2</i>	35
Gambar 3.29 Hasil Penentuan Jalur Penyelesaian Pada Titik <i>F1</i>	36
Gambar 3.30 Pendefinisian <i>Finish</i>	37
Gambar 3.31 Hasil Pendefinisian Nilai Bintang	37
Gambar 3.32 Hasil Pendefinisian Nilai <i>Start</i> dan <i>Finish</i>	38

Gambar 3.33 Daftar Substitusi Angka dengan Objek Permainan.....	39
Gambar 4.1 Tampilan Menu Utama	42
Gambar 4.2 Tampilan Pemilihan Tingkat.....	42
Gambar 4.3 Tampilan Bermain Permainan	43
Gambar 4.4 Tampilan <i>Instruction</i>	44
Gambar 4.5 Tampilan <i>Credit</i>	44
Gambar 5.1 Sistem Menggerakkan <i>Block Start</i>	54
Gambar 5.2 Pemain Mengambil Bintang	55
Gambar 5.3 Sistem Melakukan <i>Restart</i> Skenario	55
Gambar 5.4 Tampilan Pengujian Kasus Penggunaan Memilih Tingkat Permainan	59
Gambar 5.5 Tampilan Pengujian Kasus Penggunaan Bermain Permainan.....	59
Gambar 5.6 Tampilan Pengujian Kasus Penggunaan Mengulangi Tingkat Skenario	60
Gambar 5.7 Tampilan Pengujian Kasus Penggunaan Melihat Instruksi Permainan.....	61
Gambar 5.8 Tampilan Pengujian Kasus Penggunaan Melihat <i>Credit</i> Permainan.....	61

DAFTAR TABEL

Tabel 3.1 Skenario Kasus Penggunaan	15
Tabel 3.2 Skenario Kasus Penggunaan Memilih Jenis Tingkat.....	15
Tabel 3.3 Skenario Kasus Penggunaan Bermain Permainan	16
Tabel 3.4 Skenario Kasus Penggunaan Mengulangi Tingkat Skenario	16
Tabel 3.5 Skenario Kasus Penggunaan Melihat Instruksi Permainan	17
Tabel 3.6 Skenario Kasus Penggunaan Melihat <i>Credit</i> Permainan .	17
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak[6]	41
Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak[6]	51
Tabel 5.2 Skenario Pengujian Pembangkitan <i>Puzzle</i>	52
Tabel 5.3 Hasil Uji Coba Pembangkitan <i>Puzzle</i>	52
Tabel 5.4 Skenario Pengujian Aturan Main	53
Tabel 5.5 Hasil Pengujian Penyelesaian Skenario	56
Tabel 5.6 Skenario Pengujian Fungsionalitas	58
Tabel 5.7 Hasil Pengujian Fungsionalitas	62
Tabel 5.8 Daftar Nama Penguji Perangkat Lunak	63
Tabel 5.9 Penilaian Antarmuka.....	64
Tabel 5.10 Penilaian Kedinamisan Permainan.....	65
Tabel 5.11 Penilaian Kenyamanan.....	66

(Halaman Ini Sengaja Dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Inisialisasi Matriks Tingkat	45
Kode Sumber 4.2 Mengulangi Tingkat Skenario	46
Kode Sumber 4.3 Melihat Instruksi Permainan	46
Kode Sumber 4.4 Melihat <i>Credit</i> Permainan	46
Kode Sumber 4.5 <i>Midpoint Displacement</i>	47
Kode Sumber 4.6 Implementasi Penentuan Jalur	49

(Halaman Ini Sengaja Dikosongkan)

BAB I

PENDAHULUAN

Bab ini menjelaskan tentang hal-hal yang mendasari penyusunan tugas akhir yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan tugas akhir. Dengan adanya penjelasan tersebut, diharapkan tugas akhir ini dapat dipahami dengan baik.

1.1 Latar Belakang

Peran permainan sebagai sarana hiburan memang sudah tidak perlu diragukan lagi. Berbagai usia dan kalangan menjadikan permainan sebagai pelarian dari kejenuhan yang mereka hadapi. Terdapat banyak sekali jenis permainan yang bisa dijumpai saat ini. Salah satunya adalah permainan *puzzle*. Permainan *puzzle* mempunyai penggemar tersendiri dikarenakan sifatnya yang unik dan menarik.

Salah satu komponen penting dalam permainan *puzzle* adalah skenario. Skenario di setiap permainan *puzzle* mempunyai sifat yang unik dan khas. Hal inilah yang membedakan antara permainan *puzzle* yang satu dengan permainan *puzzle* lainnya. Namun, seringkali skenario yang terdapat dalam permainan *puzzle* bersifat statis. Skenario yang statis mengakibatkan kejenuhan bagi pengguna. Selain itu, skenario yang statis juga tidak memberikan tantangan yang berarti dikarenakan pengguna disajikan skenario yang sama secara berulang-ulang. Sehingga, untuk menghindari masalah seperti itu, diperlukan sebuah permainan *puzzle* yang dapat menyajikan skenario secara dinamis.

Tugas akhir ini merancang dan membangun sebuah permainan *Bangjo Puzzle* dengan skenario yang dinamis. *Midpoint Displacement* digunakan untuk menghasilkan skenario yang berubah-ubah. Selanjutnya dilakukan penentuan jalur menggunakan algoritma *Brute Force*. Skenario yang telah selesai diolah ditampilkan kepada pengguna.

1.2 Rumusan Masalah

Rumusan masalah yang diajukan dalam tugas akhir ini dijabarkan dalam poin-poin di bawah ini:

1. Bagaimana menentukan aturan main dan skenario dasar dalam permainan?
2. Bagaimana skenario dinamis diterapkan dalam permainan?
3. Bagaimana memastikan skenario yang telah dihasilkan dapat diselesaikan ?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan yang dijabarkan dalam poin-poin di bawah ini:

1. Tingkat yang ada dalam permainan tersaji dalam 3 tingkat, yaitu *easy*, *medium*, dan *hard*. Perbedaan masing-masing tingkat terdapat pada matriks yang dimiliki, yaitu 8×8 , 10×10 , dan 12×12 .
2. Bagian yang dilakukan pengacakan pada permainan ini adalah *obstacle*, posisi *start*, posisi *finish*, bintang, dan jalur.

1.4 Tujuan

Tujuan dari penyusunan tugas akhir ini adalah membangun sebuah permainan *puzzle* yang dapat menyajikan skenario yang dinamis, sehingga memberikan tantangan dan suasana yang tidak membosankan bagi penggunaanya.

1.5 Manfaat

Manfaat yang didapatkan dari penyusunan tugas akhir ini antara lain:

1. Meningkatkan tingkat kreatifitas pengguna dalam mencari setiap solusi.
2. Meningkatkan pemikiran strategis dan cermat kepada pengguna.
3. Memberikan efek hiburan kepada pengguna.
4. Memberikan suasana baru dalam bermain permainan *puzzle*.

1.6 Metodologi

Penyusunan tugas akhir ini menggunakan metodologi yang akan dijabarkan dalam poin-poin berikut ini:

A. Penyusunan Proposal Tugas Akhir

Penyusunan proposal diperlukan untuk menyampaikan gagasan dan gambaran umum tentang permainan yang akan dibuat. Selanjutnya proposal akan dijadikan acuan dalam penyusunan tugas akhir.

B. Studi Literatur

Tahap ini adalah tahap pencarian, pengumpulan, penyaringan, pemahaman, dan pembelajaran literatur terhadap segala sesuatu yang berhubungan dengan permainan *Bangjo Puzzle*. Adapun literatur-literatur yang berhubungan dengan permainan *Bangjo Puzzle* adalah sebagai berikut

1. *Game Maker*;
2. Metode pengacakan *Midpoint Displacement*;
3. Algoritma *Brute Force*;

C. Perancangan Perangkat Lunak

Tahap ini adalah lanjutan dari tahapan studi literatur. Proses perancangan sistem didasarkan kepada hasil studi literatur yang telah dilakukan pada tahapan sebelumnya. Perancangan sistem penting dilakukan agar mendapatkan hasil implementasi yang terstruktur. Proses dimulai dari perancangan komponen-komponen yang digunakan dalam membangun permainan *Bangjo Puzzle*. Selanjutnya dilakukan analisa-analisa tentang kebutuhan *non-fungsional*.

D. Implementasi dan Pembuatan Sistem

Hasil rancangan yang telah didapatkan selanjutnya diterapkan dalam bahasa pemrograman untuk mendapatkan permainan *Bangjo Puzzle* seperti yang telah dirancang pada tahapan sebelumnya.

E. Uji Coba dan Evaluasi

Tahap uji coba dan evaluasi adalah tahap selanjutnya yang dikerjakan setelah dilakukan tahapan implementasi dan pembuatan sistem. Permainan yang telah selesai dikerjakan akan dilakukan pengujian dengan mengacu pada proposal tugas akhir dan perancangan yang telah dibuat sebelumnya.

F. Penyusunan Laporan Tugas Akhir

Tahap ini adalah tahap terakhir dalam melakukan penyusunan tugas akhir. Penyusunan laporan bertujuan untuk mencatat setiap hasil yang didapatkan dalam pembangunan permainan yang berguna untuk pemeliharaan dan pengembangan di masa depan.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Laporan tugas akhir berfungsi untuk mencatat setiap hasil yang didapatkan agar terdokumentasi dengan baik. Hal ini berguna untuk pemeliharaan dan pengembangan permainan ini di masa depan. Berikut adalah sistematika penulisan laporan tugas akhir ini.

1. Bab I. Pendahuluan

Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan laporan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab ini membahas tentang pustaka penunjang yang berhubungan dengan komponen-komponen permainan *Bangjo Puzzle*.

3. Bab III. Analisis dan Perancangan

Bab ini menganalisa kebutuhan-kebutuhan yang diperlukan dalam pembuatan permainan *Bangjo Puzzle*. Selain itu, pada bab ini juga dibahas mengenai perancangan yang diperlukan dalam membangun permainan *Bangjo Puzzle*.

4. Bab IV. Implementasi

Bab ini membahas tentang implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Pada bab ini akan dijelaskan proses pembangunan permainan *Bangjo Puzzle*.

5. Bab V. Pengujian dan Evaluasi

Bab ini membahas tentang langkah-langkah pengujian permainan berikut dengan hasil yang didapatkan. Hasil tersebut selanjutnya digunakan sebagai evaluasi dalam merancang permainan *Bangjo Puzzle*.

6. Bab VI. Penutup

Bab ini berisi kesimpulan dari hasil pembangunan dan pengujian yang dilakukan serta saran untuk pengembangan permainan selanjutnya.

(Halaman Ini Sengaja Dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini akan membahas tentang teori-teori yang digunakan dalam penyusunan tugas akhir. Teori-teori tersebut yaitu pengertian *Game Maker* sebagai *game engine*, penggunaan metode *Midpoint Displacement*, dan penggunaan algoritma *Brute Force* sebagai algoritma pencari jalur.

2.1 *Game Maker*

Game Maker adalah salah satu *game engine* yang banyak digunakan dalam pembuatan permainan *desktop* maupun *mobile*. *Game Maker* dikembangkan oleh sebuah perusahaan bernama YoYo Games yang berbasis di Dundee, Skotlandia[1].

Game Maker merupakan *game engine* yang memiliki beberapa kelebihan. Hal ini menjadikan banyak pengembang permainan menggunakan *Game Maker* dalam mengembangkan permainan mereka. Berikut dijabarkan beberapa kelebihan dari *Game Maker*[2].

1. *Multiplatform Workflow*

Game Maker dapat mengekstrak aplikasi dalam berbagai macam *platform*, antara lain: *Windows Desktop*, *Mac OS*, *Ubuntu*, *Android*, *iOS*, dan masih banyak lagi.

2. *Drag and Drop*

Fitur *drag and drop* merupakan salah satu kelebihan tersendiri yang dimiliki oleh *Game Maker*. Fitur ini sangat berguna untuk mengakomodasi pengguna baru dalam *men-development* permainannya.

3. *Library yang Lengkap*

Game Maker menyediakan berbagai macam *library* yang cukup lengkap. Hal ini akan memudahkan pengguna dalam *men-development* permainan yang membutuhkan spesifikasi tinggi.

4. *GameMaker Language*

Bahasa *GameMaker* menjadi lebih mudah dipelajari karena pengembangannya didasarkan pada bahasa C. Hal ini tentu menjadi sebuah keunikan sendiri bagi sebuah *game engine*.

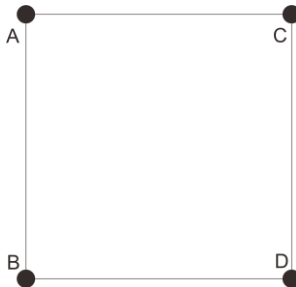
5. *Inheritance Features*

Fitur ini memungkinkan pengguna untuk melakukan pewarisan kelas-kelas dalam pemrograman. Fitur *inheritance* menjadikan program lebih terstruktur dan mudah untuk dipahami.

Saat ini terdapat dua versi yang telah dirilis oleh *Game Maker*, yaitu *Game Maker Studio 1.4* dan *Game Maker Studio 2*[3]. Penjelasan lebih detail tentang versi rilis *Game Maker* dapat dilihat dalam daftar pustaka dengan nomor urut tiga.

2.2 *Midpoint Displacement*

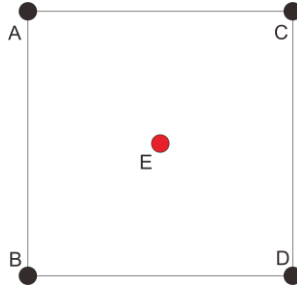
Midpoint Displacement adalah sebuah algoritma pembagian[4]. Algoritma ini digunakan untuk pembuatan *random map* dengan cara melakukan *generate* angka pada matriks tertentu. Angka-angka tersebut dicari nilai tengahnya kemudian ditambah dengan nilai acak agar bisa terbentuk angka baru. Angka baru yang telah terbentuk akan diproses ulang hingga matriks terisi penuh dengan angka-angka. Selanjutnya angka tersebut disubstitusi dengan objek-objek penyusun *map*. Metode *Midpoint Displacement* yang digunakan dalam permainan *Bangjo Puzzle* akan diterapkan dalam matriks dua dimensi. Proses dari *Midpoint Displacement* akan dijabarkan dalam beberapa gambar di bawah ini.



Gambar 2.1 Proses Awal *Midpoint Displacement*

Pada Gambar 2.1, hal yang pertama dilakukan adalah melakukan *generate* angka secara acak pada titik hitam, yaitu *A*, *B*, *C*, dan *D*. Langkah selanjutnya adalah melakukan *generate* pada titik tengah *E* dengan menggunakan Persamaan 2.1 yang ditunjukkan pada Gambar 2.2.

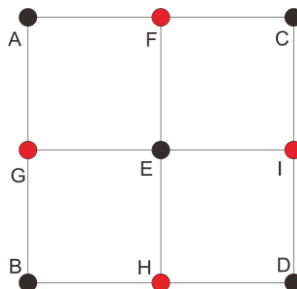
$$E = (A+B+C+D) / 4 + \text{random value} \quad (2.1)$$



Gambar 2.2 Proses Mencari Nilai Pusat

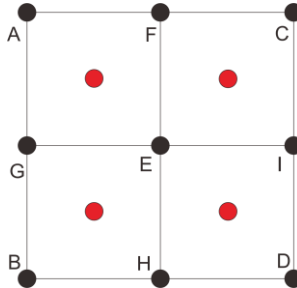
Langkah selanjutnya adalah melakukan *generate* pada titik *F*, *G*, *H*, dan *I*. Persamaan yang digunakan dalam mencari nilai pada titik tersebut adalah Persamaan 2.2. Ilustrasi yang menggambarkan proses ini ditunjukkan pada Gambar 2.3.

$$\begin{aligned} F &= (A+C) / 2 + \text{random value} \\ G &= (A+B) / 2 + \text{random value} \\ H &= (B+D) / 2 + \text{random value} \\ I &= (C+D) / 2 + \text{random value} \end{aligned} \quad (2.2)$$



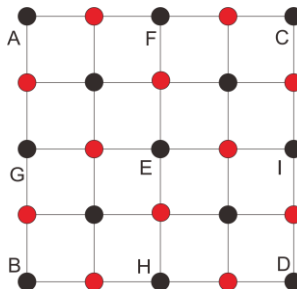
Gambar 2.3 Proses Mencari Nilai Tengah

Langkah selanjutnya adalah mencari nilai pusat dari persegi A,G,E,F ; persegi F,E,C,I ; persegi G,B,E,H ; dan persegi E,H,I,D . Persamaan 2.1 digunakan untuk mencari keempat titik tersebut. Proses ini ditunjukkan pada Gambar 2.4.



Gambar 2.4 Proses Mencari Nilai Pusat Pada Keempat Persegi

Langkah selanjutnya adalah mencari nilai tengah pada masing-masing titik A,F ; titik F,C ; titik A,G ; titik F,E ; dst. Sehingga didapatkan hasil seperti yang ditunjukkan pada Gambar 2.5. Persamaan 2.2 digunakan untuk mencari nilai pada titik-titik ini.



Gambar 2.5 Proses Mencari Nilai Tengah Beberapa Titik

Langkah ini akan terus berulang hingga semua matriks terisi dengan angka. Proses selanjutnya adalah melakukan penentuan jalur agar *map* yang telah dihasilkan dapat terselesaikan oleh pemain. Metode yang digunakan dalam melakukan penentuan jalur adalah algoritma *Brute Force*.

2.3 Algoritma *Brute Force*

Algoritma *Brute Force* merupakan suatu metode yang digunakan untuk memecahkan masalah dengan memperhatikan konsep permasalahan secara langsung[5]. Algoritma ini menyelesaikan permasalahan secara sederhana dengan cara yang cukup jelas. Kelebihan lainnya yang dimiliki oleh algoritma *Brute Force* dijabarkan dalam beberapa poin berikut ini.

1. *Straightforward*

Algoritma *Brute Force* menyelesaikan permasalahan secara langsung dengan berdasarkan pada pernyataan masalah dan konsep yang ada pada permasalahan tersebut.

2. Digunakan Untuk Berbagai Permasalahan

Algoritma *Brute Force* digunakan untuk memecahkan berbagai macam masalah seperti, pencarian, pengurutan, pencocokan *string*, dan perkalian matriks.

3. Menghasilkan Algoritma Baku

Algoritma *Brute Force* menghasilkan algoritma baku pada permasalahan komputasi seperti penjumlahan atau perkalian n bilangan dan penentuan nilai minimum dan maksimum suatu deret bilangan.

4. Mudah Digunakan

Algoritma *Brute Force* merupakan algoritma yang mudah digunakan. Sifatnya yang langsung berfokus pada permasalahan membuat algoritma ini memiliki kemudahan dalam penerapannya.

(Halaman Ini Sengaja Dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Bab ini akan membahas mengenai analisis dan perancangan sistem permainan *Bangjo Puzzle*.

3.1 Analisis Sistem

Bab ini akan membahas mengenai analisis sistem dari permainan *Bangjo Puzzle*. Komponen-komponen yang dilakukan analisis adalah kebutuhan fungsional sistem, kebutuhan non-fungsional, deskripsi umum perangkat lunak, dan karakteristik pengguna.

3.1.1 Analisis Kebutuhan Fungsional

Pada pembangunan permainan *Bangjo Puzzle*, terdapat beberapa kebutuhan yang akan dikerjakan. Kebutuhan-kebutuhan tersebut akan dijabarkan dalam poin-poin di bawah ini.

a. Memilih Tingkat Permainan

Pengguna dapat memilih tiga kelompok tingkat yang akan dimainkan, yaitu *easy*, *medium*, dan *hard*. Setiap tingkat memiliki spesifikasi waktu dan jumlah papan yang berbeda.

b. Mengulangi Tingkat Skenario

Pemain dapat melakukan pengulangan skenario apabila melakukan kesalahan gerakan sehingga terjadi *deadlock*.

c. Melihat Instruksi Permainan

Pemain dapat melihat instruksi permainan. Instruksi permainan berisi tentang informasi permainan.

d. Melihat *Credit* Permainan

Pemain dapat melihat *credit* permainan. *Credit* permainan dapat dilihat pada halaman *credit*.

3.1.2 Deskripsi Umum Perangkat Lunak

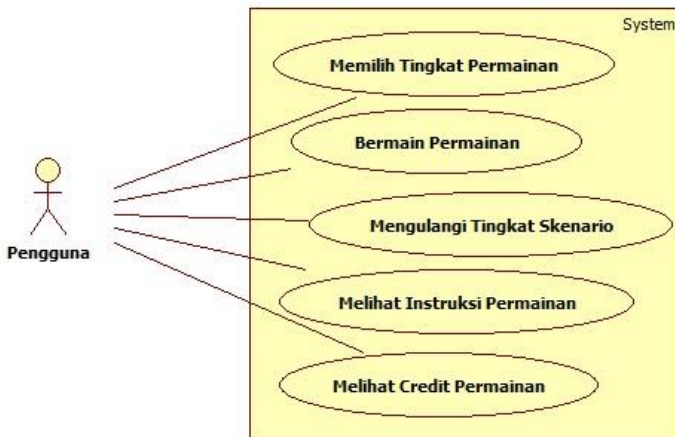
Permainan *Bangjo Puzzle* adalah sebuah permainan *puzzle* yang mengharuskan pengguna memindahkan *block* dari posisi awal menuju posisi akhir dengan sebelumnya mengambil bintang terlebih dahulu. Terdapat 3 macam tingkat, yaitu *easy*, *medium*, dan *hard*. Masing-masing dari tingkat mempunyai skenario, dan papan yang berbeda-beda.

Permainan *Bangjo Puzzle* dibangun menggunakan *game engine Game Maker*. Platform yang digunakan dalam permainan ini adalah Android. Penggunaan Android sebagai *platform* diharapkan dapat mempermudah pengguna dalam menggunakan permainan ini.

3.2 Perancangan Sistem

3.2.1 Perancangan Fungsionalitas Aplikasi

Berikut akan ditunjukkan diagram fungsionalitas permainan *Bangjo Puzzle* yang tersaji pada Gambar 3.1.



Gambar 3.1 Diagram Fungsionalitas Permainan

3.2.2 Perancangan Skenario Kasus Penggunaan

Berdasarkan Gambar 3.1, skenario kasus penggunaan permainan *Bangjo Puzzle* akan dijabarkan dalam Tabel 3.1.

Tabel 3.1 Skenario Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memilih Tingkat Permainan	Pengguna dapat menentukan jenis tingkat permainan yang akan dimainkan.
2	UC-002	Bermain Permainan	Pengguna melakukan kegiatan bermain permainan pada skenario tertentu.
3	UC-003	Mengulangi Tingkat Skenario	Pengguna melakukan pengulangan skenario terakhir yang dihasilkan oleh komputer.
4	UC-004	Melihat Instruksi Permainan	Pengguna dapat mempelajari cara bermain permainan <i>Bangjo Puzzle</i> .
5	UC-005	Melihat <i>Credit</i> Permainan	Pengguna dapat melihat <i>credit</i> permainan <i>Bangjo Puzzle</i> pada menu yang telah disediakan.

3.2.2.1 Kasus Penggunaan Permainan

Penjelasan kasus penggunaan permainan untuk skenario UC-001, yakni memilih jenis tingkat akan dijabarkan dalam Tabel 3.2.

Tabel 3.2 Skenario Kasus Penggunaan Memilih Jenis Tingkat

Nama Kasus Penggunaan	Memilih Tingkat Permainan
Kode	UC-001
Deskripsi	Pengguna melakukan pemilihan tingkat permainan yang akan dimainkan.
Aktor	Pemain.

Kondisi Awal	Pemain menjalankan permainan <i>Bangjo Puzzle</i> hingga muncul tampilan utama. Kemudian pemain memilih menu <i>Play Game</i> .
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih menu <i>Play Game</i>. 2. Sistem menampilkan pilihan tingkat. 3. Pemain memilih salah satu dari tingkat yang ditampilkan komputer. 4. Sistem menampilkan skenario sesuai dengan spesifikasi tingkat-nya.

Penjelasan kasus penggunaan permainan untuk skenario UC-002 yakni bermain permainan dijelaskan pada Tabel 3.3.

Tabel 3.3 Skenario Kasus Penggunaan Bermain Permainan

Nama Kasus Penggunaan	Bermain Permainan
Kode	UC-002
Deskripsi	Pengguna melakukan kegiatan bermain permainan pada tingkat yang telah dipilih sebelumnya.
Aktor	Pemain.
Kondisi Awal	Pemain telah memilih jenis tingkat yang akan dimainkan pada menu <i>Play Game</i> .
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih jenis tingkat. 2. Sistem menghasilkan dan menampilkan skenario sesuai dengan tingkat yang dipilih. 3. Pemain menyelesaikan skenario. 4. Sistem menghasilkan skenario baru.

Penjelasan kasus penggunaan permainan untuk skenario UC-003 yakni mengulangi tingkat skenario dijelaskan pada Tabel 3.4.

Tabel 3.4 Skenario Kasus Penggunaan Mengulangi Tingkat Skenario

Nama Kasus Penggunaan	Mengulangi Tingkat Skenario
Kode	UC-003
Deskripsi	Pengguna melakukan pengulangan skenario yang gagal diselesaikan.

Aktor	Pemain.
Kondisi Awal	Pemain sudah dalam proses memainkan permainan dan mengalami buntu dalam pergerakan.
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih tombol ikon untuk <i>restart</i>. 2. Sistem melakukan pengulangan posisi komponen permainan pada skenario tersebut.

Penjelasan kasus penggunaan permainan untuk skenario UC-004 yakni melihat instruksi permainan dijelaskan pada Tabel 3.5.

Tabel 3.5 Skenario Kasus Penggunaan Melihat Instruksi Permainan

Nama Kasus Penggunaan	Melihat Instruksi Permainan
Kode	UC-004
Deskripsi	Pengguna mempelajari instruksi dalam menggunakan aplikasi.
Aktor	Pemain.
Kondisi Awal	Pemain menjalankan permainan <i>Bangjo Puzzle</i> hingga muncul tampilan utama. Kemudian pemain memilih menu <i>Instruction</i> .
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih menu <i>Instruction</i>. 2. Sistem menampilkan instruksi yang diperlukan dalam memainkan permainan <i>Bangjo Puzzle</i>.

Penjelasan kasus penggunaan permainan untuk skenario UC-005 yakni melihat *credit* permainan dijelaskan pada Tabel 3.6.

Tabel 3.6 Skenario Kasus Penggunaan Melihat *Credit* Permainan

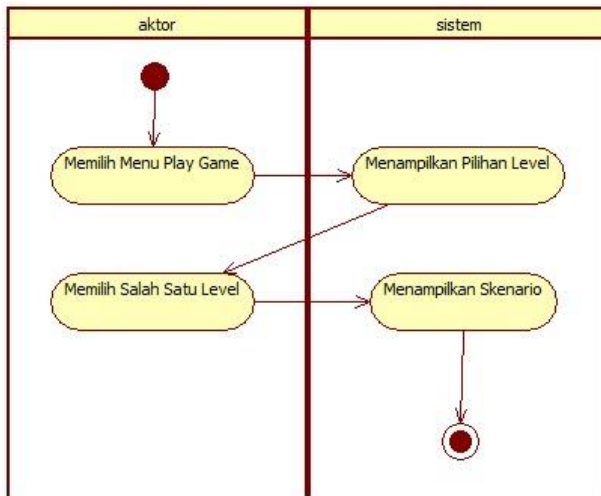
Nama Kasus Penggunaan	Melihat <i>Credit</i> Permainan
Kode	UC-005
Deskripsi	Pengguna melihat <i>credit</i> yang ada di dalam permainan.
Aktor	Pemain.
Kondisi Awal	Pemain menjalankan permainan <i>Bangjo Puzzle</i> hingga muncul tampilan utama. Kemudian pemain memilih menu <i>Credit</i> .

Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih menu <i>Credit</i>. 2. Sistem menampilkan <i>credit</i> permainan <i>Bangjo Puzzle</i>.
--------------------	--

3.2.2.2 Diagram Aktivitas

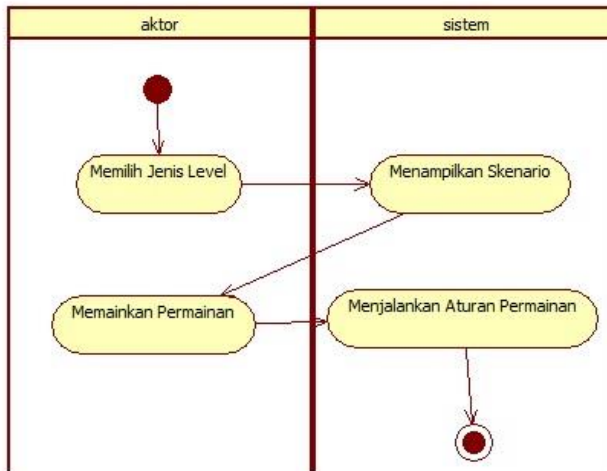
Berikut ini akan disajikan diagram aktivitas untuk masing-masing skenario kasus penggunaan yang telah dijelaskan pada Tabel 3.2 sampai dengan Tabel 3.6.

Diagram aktivitas dari kasus penggunaan UC-001 yakni memilih tingkat permainan dijelaskan pada Gambar 3.2.



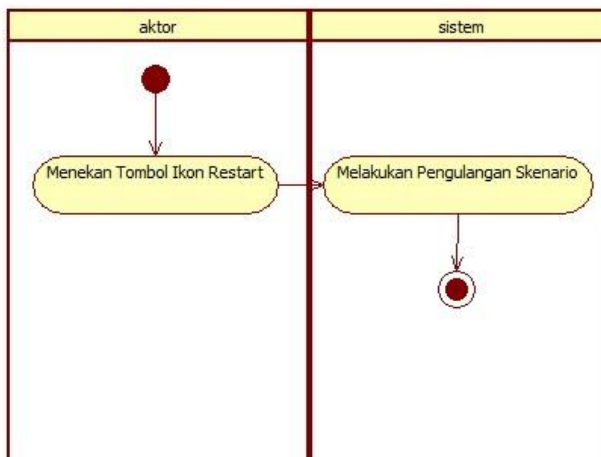
Gambar 3.2 Diagram Aktivitas Memilih Tingkat Permainan

Diagram aktivitas dari kasus penggunaan UC-002 yakni bermain permainan dijelaskan pada Gambar 3.3.



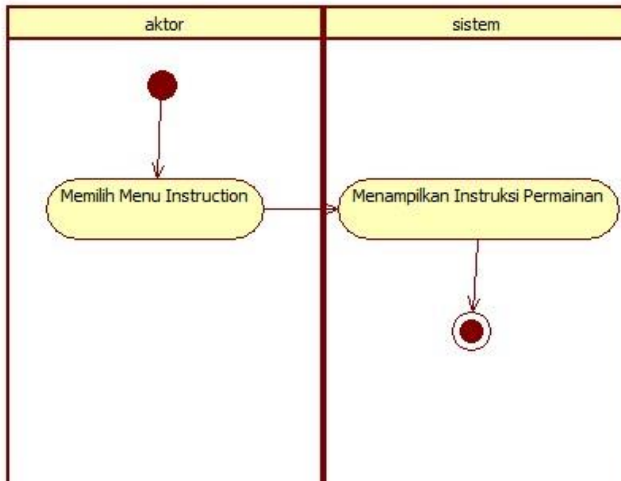
Gambar 3.3 Diagram Aktivitas Bermain Permainan

Diagram aktivitas dari kasus penggunaan UC-003 yakni mengulangi tingkat skenario dijelaskan pada Gambar 3.4.



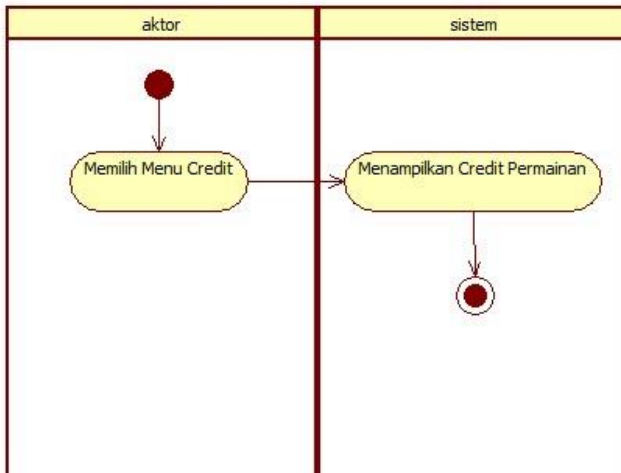
Gambar 3.4 Diagram Aktivitas Mengulangi Tingkat Skenario

Diagram aktivitas dari kasus penggunaan UC-004 yakni melihat instruksi permainan dijelaskan pada Gambar 3.5.



Gambar 3.5 Diagram Aktivitas Melihat Instruksi Permainan

Diagram aktivitas dari kasus penggunaan UC-005 yakni melihat *credit* permainan dijelaskan pada Gambar 3.6.



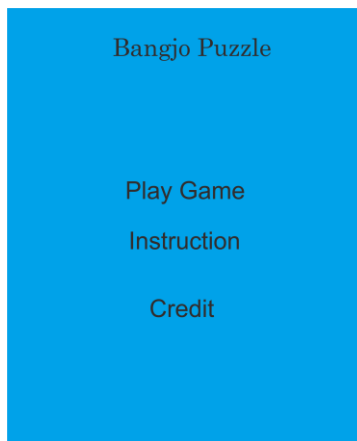
Gambar 3.6 Diagram Aktivitas Melihat *Credit* Permainan

3.2.3 Perancangan Antarmuka Pengguna

Subbab ini membahas rancangan antarmuka yang digunakan dalam permainan *Bangjo Puzzle*. Terdapat beberapa tampilan antarmuka yang selanjutnya dijabarkan dalam subbagian di bawah ini.

3.2.3.1 Tampilan Menu Utama

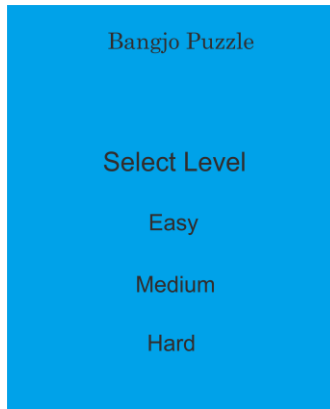
Tampilan Menu Utama adalah tampilan yang ditampilkan pertama kali terlihat saat pemain menjalankan aplikasi. Terdapat tiga menu utama yang disajikan dalam permainan ini, yaitu *Play Game*, *Instruction*, dan *Credits*. Terdapat juga tombol *close* untuk keluar dari aplikasi. Perancangan tampilan menu utama ditunjukkan pada Gambar 3.7.



Gambar 3.7 Perancangan Tampilan Menu Utama

3.2.3.2 Tampilan Pemilihan Tingkat

Tampilan ini adalah tampilan yang bertujuan untuk memilih tingkat sebelum permainan dimulai. Terdapat tiga macam tingkatan tingkat, yaitu *easy*, *medium*, dan *hard*. Perancangan tampilan pemilihan tingkat ditunjukkan pada Gambar 3.8.



Gambar 3.8 Perancangan Tampilan Pemilihan Tingkat

3.2.3.3 Tampilan Bermain Permainan

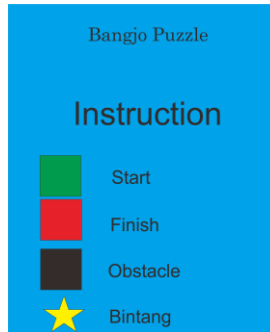
Tampilan ini adalah tampilan utama dalam permainan *Bangjo Puzzle*. Tampilan ini ditampilkan setelah pemain memilih tingkat pada tampilan pemilihan tingkat. Pemain diharuskan memindahkan *block* berwarna merah menuju *block* berwarna hijau. Pemain diwajibkan untuk mengambil bintang agar bisa memenangkan permainan. Perancangan tampilan bermain permainan ditunjukkan pada Gambar 3.9.



Gambar 3.9 Perancangan Tampilan Bermain Permainan

3.2.3.4 Tampilan *Instruction*

Tampilan ini adalah tampilan yang bertujuan untuk membantu pengguna dalam memainkan permainan *Bangjo Puzzle*. Terdapat penjelasan singkat mengenai tujuan dari permainan dan beberapa komponen penting lainnya. Perancangan tampilan *instruction* ditunjukkan pada Gambar 3.10.



Gambar 3.10 Perancangan Tampilan *Instruction*

3.2.3.5 Tampilan *Credit*

Tampilan *credit* adalah tampilan yang bertujuan untuk menampilkan kredit yang diberikan kepada semua pihak yang telah membantu terwujudnya permainan ini. Perancangan tampilan *credit* ditunjukkan pada Gambar 3.11.



Gambar 3.11 Perancangan Tampilan *Credit*

3.2.4 Perancangan Kontrol Permainan

Kontrol permainan yang digunakan dalam permainan ini adalah menggunakan *joystick* dan ketuk layar. *Joystick* ini terdiri dari lima kontrol ikon, yaitu atas, bawah, kanan, kiri, dan *restart*. Fungsi ikon atas, bawah, kanan, dan kiri adalah untuk memindahkan *block* yang satu menuju *block* yang lainnya. Ikon *restart* digunakan untuk mengulangi skenario terakhir yang sedang dijalankan. Sedangkan fungsi ketuk adalah untuk memilih pilihan-pilihan sesuai dengan fungsionalitasnya.

3.2.5 Perancangan Aturan Permainan

Aturan permainan adalah salah satu komponen penting yang ada dalam permainan. Aturan permainan adalah suatu tata cara yang diperlukan untuk menyelesaikan sebuah permainan. Terdapat beberapa aturan dasar dalam permainan *Bangjo Puzzle* yang akan dijabarkan dalam beberapa poin berikut ini.

- a. *Block start* hanya dapat bergerak secara lurus. Pergerakan *block start* hanya akan berhenti apabila telah menabrak *obstacle* atau berada di ujung papan.
- b. Untuk menyelesaikan skenario, pemain diharuskan mengambil seluruh bintang. Permainan tidak akan selesai apabila pemain masih menyisakan bintang.
- c. Skenario akan sukses terselesaikan apabila pemain berhasil mengambil seluruh bintang dan memindahkan *block start* menuju *block finish*.
- d. Pemain diperkenankan mengulangi skenario apabila terjadi kesalahan gerakan yang mengakibatkan posisi *block* menjadi *deadlock*.

3.2.6 Perancangan Metode

Permainan *Bangjo Puzzle* menggunakan metode yang digunakan sebagai pendukung komponen permainan. Metode yang digunakan dalam permainan *Bangjo Puzzle* terbagi menjadi dua, yaitu metode

yang digunakan untuk mengacak *puzzle map* dan menentukan jalur *puzzle*.

3.2.6.1 Perancangan Data *Puzzle Map*

Puzzle map adalah sebuah tampilan yang menggambarkan papan permainan beserta beberapa komponennya. *Puzzle map* dihasilkan secara otomatis oleh komputer menggunakan metode *Midpoint Displacement* seperti yang telah dijelaskan pada Bab 2. Berikut akan dijabarkan proses komputer melakukan metode *Midpoint Displacement*.

3.2.6.1.1 Pendefinisian Objek

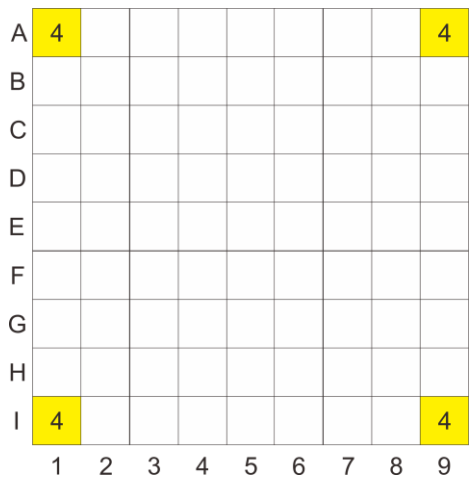
Terdapat beberapa objek yang menjadi komponen dari *puzzle map*. Objek-objek tersebut yaitu *obstacle*, bintang, *block start*, dan *block finish*. Gambar 3.12 adalah daftar objek yang terdapat dalam permainan *Bangjo Puzzle*.



Gambar 3.12 Daftar Objek Permainan *Bangjo Puzzle*

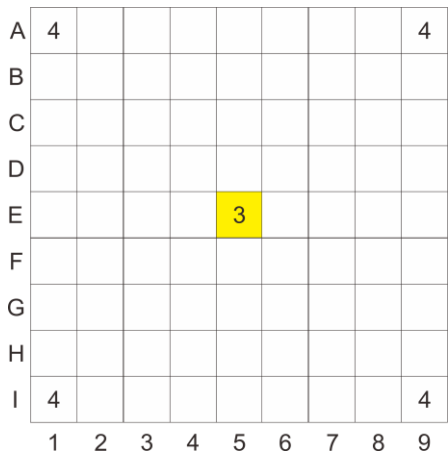
3.2.6.1.2 *Midpoint Displacement*

Jumlah matriks yang digunakan dalam ilustrasi di bawah ini adalah 9x9. Posisi matriks yang pertama kali diberikan nilai adalah pada empat titik masing-masing penjuru. *Initial value* yang dikirimkan kepada titik penjuru adalah empat. Pemilihan angka empat bertujuan untuk memberikan kemudahan pada saat dimasukkan ke dalam persamaan dan menjadikan masing-masing penjuru menjadi *obstacle* pada proses terakhir. Gambar 3.13 adalah tahap awal proses *Midpoint Displacement*.



Gambar 3.13 *Midpoint Displacement* (Langkah 1)

Proses selanjutnya yang dilakukan adalah mencari titik tengah yang terletak pada matriks *E5*. Persamaan yang digunakan untuk mencari nilai pada titik tersebut merujuk pada Persamaan 2.1 pada Bab 2. Gambar 3.14 mempresentasikan hasil yang didapatkan pada proses ini.



Gambar 3.14 *Midpoint Displacement* (Langkah 2)

Langkah selanjutnya adalah mencari nilai pada titik $A5$, $E1$, $I5$, dan $E9$. Persamaan yang digunakan untuk mencari nilai pada titik tersebut merujuk pada Persamaan 2.2 pada Bab 2. Gambar 3.15 mempresentasikan hasil yang didapatkan pada proses ini.

A	4			2				4	
B									
C									
D									
E	1			3				0	
F									
G									
H									
I	4			0				4	
	1	2	3	4	5	6	7	8	9

Gambar 3.15 Midpoint Displacement (Langkah 3)

Gambar di atas memperlihatkan bahwa telah terbentuk persegi baru, yaitu $A1, E1, A5, E5$; $A5, E5, A9, E9$; $E1, I1, E5, I5$; dan $E5, I5, E9, I9$. Pada masing-masing matriks tersebut akan dilakukan pengulangan proses *Midpoint Displacement* seperti pada langkah pertama. Gambar 3.16 sampai dengan Gambar 3.18 mempresentasikan proses pengulangan yang dilakukan sampai dengan mendapatkan hasilnya.

A	4				2				4
B									
C			0				0		
D									
E	1				3				0
F									
G			1				4		
H									
I	4				0				4
	1	2	3	4	5	6	7	8	9

Gambar 3.16 Midpoint Displacement (Langkah 4)

A	4		0		2		0		4
B									
C	3		0		1		0		3
D									
E	1		0		3		2		0
F									
G	0		1		0		4		3
H									
I	4		0		0		1		4
	1	2	3	4	5	6	7	8	9

Gambar 3.17 *Midpoint Displacement* (Langkah 5)

A	4	0	0	0	2	2	0	1	4
B	2	2	0	1	4	3	4	4	4
C	3	0	0	0	1	0	0	0	3
D	0	4	3	4	1	4	3	2	2
E	1	0	0	0	3	0	2	0	0
F	1	1	1	3	2	3	1	2	2
G	0	0	1	0	0	0	4	0	3
H	3	2	1	1	1	4	1	1	1
I	4	1	0	4	0	0	1	0	4
	1	2	3	4	5	6	7	8	9

Gambar 3.18 Hasil Proses *Midpoint Displacement*

Proses *Midpoint Displacement* telah berhasil dilakukan dengan ditunjukkan pada Gambar 3.18. Proses selanjutnya adalah memodifikasi angka-angka agar dapat digunakan dalam perancangan data jalur *puzzle*, yaitu sebuah perancangan yang bertujuan untuk mencari jalur pada *random map* agar permainan dapat terselesaikan. algoritma *Brute Force* digunakan untuk penentuan jalur *puzzle*. Proses perancangan data jalur *puzzle* akan dibahas pada Subbab 3.2.6.2.

3.2.6.2 Perancangan Data Jalur *Puzzle*

Jalur *puzzle* adalah jalur yang digunakan pemain untuk memindahkan blok *start* menuju blok *finish*. Metode yang digunakan dalam melakukan penentuan jalur adalah metode algoritma *Brute Force*. Sistem akan menentukan jalur *puzzle* dimulai dari posisi *start*. Proses pengecekan dilakukan pada setiap titik kemungkinan yang terdapat di sekitar *start*. Titik yang diambil adalah titik yang mempunyai nilai jumlah jalur paling besar. Jumlah jalur minimal yang diharuskan adalah sebanyak empat. Apabila kurang dari itu, komputer akan mengulangi proses *Midpoint Displacement* dan melakukan pengecekan lagi menggunakan algoritma *Brute Force*. Proses ini menentukan pola jalur permainan, bintang, dan posisi *block finish* agar skenario yang ditampilkan dapat diselesaikan oleh pemain.

3.2.6.2.1 Pengubahan Nilai Menjadi Nol dan Satu

Setelah proses *Midpoint Displacement* selesai dilakukan, proses selanjutnya adalah mengubah nilai dari setiap matriks menjadi nol dan satu. Tujuan pengubahan angka menjadi nol dan satu adalah untuk mempermudah proses penentuan jalur, menyederhanakan tampilan papan, dan memperbanyak jalan. Papan dikatakan kurang efektif apabila terlalu banyak *obstacle* daripada jalan. Hasil dari proses pengubahan nilai menjadi nol dan satu ditunjukkan pada Gambar 3.19.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.19 Hasil Pengubahan Angka Menjadi Nol dan Satu

Aturan yang diterapkan dalam mengubah angka nol dan satu adalah dengan cara mengubah nilai dengan rentang lebih dari tiga dan kurang dari lima menjadi satu dan mengubah nilai di luar rentang tersebut menjadi nol. Pemilihan rentang angka tiga dan lima agar didapatkan jalan lebih banyak daripada *obstacle*.

3.2.6.2.2 Penentuan Matriks *Start*

Proses selanjutnya adalah menentukan matriks posisi *start*. Proses ini sangat penting, karena proses penentuan jalur akan berjalan mulai dari posisi *start*. Proses penentuan posisi *start* ditentukan secara acak pada posisi matriks yang memiliki nilai nol (jalan). Pada Gambar 3.20 ditunjukkan bahwa titik *E1* digunakan sebagai posisi *start*.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.20 Penentuan Posisi *Start*

3.2.6.2.3 Proses Penentuan Jalur

Proses selanjutnya adalah penentuan jalur menggunakan algoritma *Brute Force*. Sistem akan memainkan permainan sesuai dengan aturan yang telah dibuat, yaitu menggerakkan posisi *start* menuju posisi *finish*. Terdapat tiga titik kemungkinan yang dapat dilalui pertama kali, yaitu menuju *D1*, *F1*, dan *E2*. Sistem akan melakukan proses penentuan jalur dan membandingkan berdasarkan langkah terbanyak. Titik yang memiliki langkah terbanyak selanjutnya dipilih sebagai titik penyelesaian beserta dengan jalur di dalamnya. Pengecekan yang dilakukan pertama kali adalah pada titik *D1*. Gambar 3.21 menunjukkan proses penentuan jalur melalui titik *D1*.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.21 Proses Penentuan Jalur Penyelesaian Melalui Titik *D1*

Sistem akan terus melakukan pengecekan ke arah atas hingga menemukan *obstacle* atau berada pada ujung papan. Sistem dikatakan sukses menghasilkan jalur apabila telah berjalan dan berhenti pada *obstacle* atau ujung papan. Jalur ini selanjutnya disebut dengan Jalur Penyelesaian. Jalur Penyelesaian pertama yang dihasilkan oleh sistem, ditunjukkan pada Gambar 3.22.

A	1	0	0	0	0	0	0	1	
B	0	0	0	0	1	0	1	1	
C	0	0	0	0	0	0	0	0	
D	0	1	0	1	0	1	0	0	
E	0	0	0	0	0	0	0	0	
F	0	0	0	0	0	0	0	0	
G	0	0	0	0	0	0	1	0	
H	0	0	0	0	0	1	0	0	
I	1	0	0	1	0	0	0	1	
	1	2	3	4	5	6	7	8	9

Gambar 3.22 Jalur Penyelesaian Pertama Pada Titik *D1*

Selanjutnya sistem akan melakukan pengecekan arah pada titik yang akan di cek. Proses pengecekan arah ini selanjutnya disebut Cek Jalur. Cek Jalur dilakukan secara acak. . Cek Jalur digunakan untuk menentukan arah pergerakan selanjutnya. Nilai Cek Jalur terus bertambah walaupun sistem tidak menemukan titik yang dapat dilalui. Dalam ilustrasi ini, Cek Jalur diilustrasikan bergerak ke arah kiri. Arah kiri merupakan tembok pembatas sehingga tidak mungkin dilakukan. Sistem kemudian melakukan pengacakan kembali dan

menghasilkan Cek Jalur ke kanan. Gambar 3.23 menunjukkan Jalur Penyelesaian kedua dan Cek Jalur kedua.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.23 Jalur Penyelesaian Kedua Pada Titik *DI*

Gambar 3.24 menunjukkan Jalur Penyelesaian Ketiga dan Cek Jalur keempat. Cek Jalur ketiga dilakukan pada sebelah kanan.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.24 Jalur Penyelesaian Ketiga Pada Titik *DI*

Gambar 3.25 menunjukkan Jalur Penyelesaian keempat dan Cek Jalur keenam. Cek Jalur kelima dilakukan pada sebelah atas.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.25 Jalur Penyelesaian Keempat Pada Titik *DI*

Gambar 3.26 menunjukkan Jalur Penyelesaian kelima dan Cek Jalur kesembilan. Cek Jalur ketujuh dilakukan pada sebelah kiri dan Cek Jalur kedelapan dilakukan pada sebelah atas.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.26 Jalur Penyelesaian Kelima Pada Titik *DI*

Proses penentuan jalur akan selesai dilakukan apabila Cek Jalur telah mencapai angka lima belas. Jalur Penyelesaian yang didapatkan pada titik *D1* adalah sebanyak delapan. Hasil penentuan jalur pada titik *D1* ditunjukkan pada Gambar 3.27.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.27 Hasil Penentuan Jalur Penyelesaian Pada Titik *D1*

Langkah selanjutnya adalah penentuan jalur pada titik *E2*. Proses penentuan jalur sama dengan proses penentuan jalur pada titik *D1*. Hasil penentuan jalur pada titik *E2* menghasilkan Jalur Penyelesaian sebanyak sepuluh. Gambar 3.28 menunjukkan hasil dari penentuan jalur pada titik *E2*.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.28 Hasil Penentuan Jalur Penyelesaian Pada Titik *E2*

Langkah selanjutnya adalah penentuan jalur penyelesaian pada titik *F1*. Proses penentuan jalur sama dengan proses penentuan jalur pada titik *E2*. Hasil penentuan jalur pada titik *F1* menghasilkan Jalur Penyelesaian sebanyak lima. Gambar 3.29 menunjukkan hasil penentuan jalur pada titik *F1*.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.29 Hasil Penentuan Jalur Penyelesaian Pada Titik F1

Langkah selanjutnya adalah melakukan pemilihan jalur pada titik yang mempunyai Jalur Penyelesaian terbanyak. Dalam ilustrasi ini, titik *E2* dipilih karena mempunyai Jalur Penyelesaian terbanyak.

3.2.6.2.4 Pendefinisian *Finish*

Matriks akhir yang didapatkan pada proses sebelumnya, akan didefinisikan sebagai posisi *finish*. Ilustrasi posisi *finish* ditunjukkan pada Gambar 3.30 yang ditandai dengan warna hijau.

A	1	0	0	0	0	0	0	0	1
B	0	0	0	0	1	0	1	1	1
C	0	0	0	0	0	0	0	0	0
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	0	0	0
I	1	0	0	1	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.30 Pendefinisian *Finish*

3.2.6.2.5 Pendefinisian Bintang

Pendefinisian bintang dilakukan ketika proses penentuan jalur sedang dilakukan, yaitu pada nilai matriks yang bernilai nol dan pada saat melakukan perubahan arah. Pendefinisian ini dilakukan pada jalur yang sedang dikerjakan. Apabila suatu matriks akan diletakkan sebuah bintang, maka nilai yang semula nol akan diubah menjadi empat. Gambar 3.31 menunjukkan hasil dari pendefinisian nilai bintang.

A	1	0	0	0	0	0	0	0	1
B	4	0	0	0	1	0	1	1	1
C	4	0	0	0	4	0	0	0	4
D	0	1	0	1	0	1	0	0	0
E	0	0	0	0	0	0	0	0	4
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	4	0	4
I	1	0	0	1	4	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.31 Hasil Pendefinisian Nilai Bintang

3.2.6.2.6 Pendefinisian Nilai *Start dan Finish*





Proses ini adalah pendefinisian nilai posisi start dan finish menjadi dua dan tiga. Pendefinisian ini bertujuan untuk memudahkan pada saat substitusi menjadi objek permainan yang selanjutnya ditampilkan kepada pengguna.. Hasil pendefinisian nilai *start* dan *finish* ditunjukkan pada Gambar 3.32.

A	1	0	0	0	0	0	0	0	1
B	4	0	0	3	1	0	1	1	1
C	4	0	0	0	4	0	0	0	4
D	0	1	0	1	0	1	0	0	0
E	2	0	0	0	0	0	0	0	4
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	1	4	0	4
I	1	0	0	1	4	0	0	0	1
	1	2	3	4	5	6	7	8	9

Gambar 3.32 Hasil Pendefinisian Nilai *Start dan Finish*

3.2.6.2.7 Pendefinisian Objek

Matriks yang sudah jadi siap untuk ditampilkan kepada pengguna. Sistem akan merubah angka-angka tersebut menjadi objek-objek penyusun permainan *Bangjo Puzzle* sebelum ditampilkan kepada pengguna. Gambar 3.33 adalah daftar angka yang akan dilakukan substitusi dengan objek permainan sebelum ditampilkan kepada pengguna.

- 1 ~~~>  *Obstacle*
- 2 ~~~>  *Block Start*
- 3 ~~~>  *Block Finish*
- 4 ~~~>  *Bintang*

Gambar 3.33 Daftar Substitusi Angka dengan Objek Permainan

(Halaman Ini Sengaja Dikosongkan)

BAB IV IMPLEMENTASI

Bab ini akan membahas implentasi sistem permainan *Bangjo Puzzle* sesuai dengan analisis dan perancangan yang telah dilakukan di Bab III. Implementasi yang akan dibahas mencakup algoritma dan rancangan antarmuka.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang digunakan dalam tugas akhir ini dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak[6]

Perangkat Keras	<i>Device</i> : Sony Xperia M2 Aqua RAM : 1 Gb Internal Memori : 8 Gb Resolusi Layar : 540 x 960 pixel
Perangkat Lunak	Sistem Operasi : Google Android 5.1 (Lollipop)

4.2 Implementasi Permainan

Implementasi yang dilakukan pada permainan *Bangjo Puzzle* mencakup aspek tampilan antarmuka, kasus penggunaan, dan metode yang digunakan dalam membangkitkan skenario acak berikut dengan penyelesaian skenario tersebut.

4.2.1 Implementasi Tampilan Menu Utama

Tampilan menu utama adalah tampilan awal yang akan terlihat oleh pengguna saat pertama kali menjalankan aplikasi. Terdapat tiga submenu yang dapat dipilih oleh pengguna, yaitu *Play Game*, *Instruction*, dan *Credit*. Ikon untuk menutup aplikasi juga tersedia

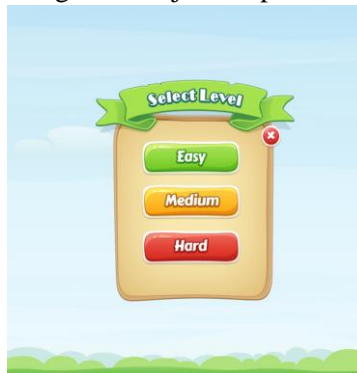
pada sisi sudut atas kanan permainan. Gambar 4.1 adalah tampilan menu utama permainan *Bangjo Puzzle*.



Gambar 4.1 Tampilan Menu Utama

4.2.2 Implementasi Tampilan Pemilihan Tingkat

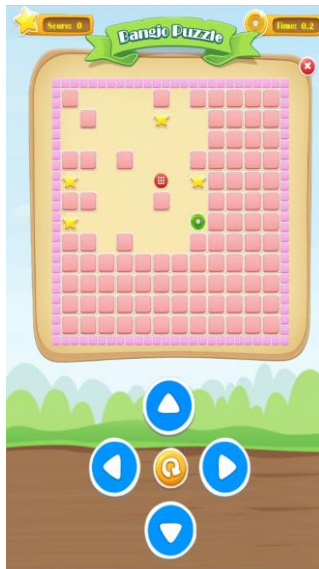
Tampilan pemilihan tingkat akan muncul ketika pengguna memilih menu *Play Game*. Terdapat tiga tingkat yang dapat dipilih oleh pengguna, yaitu *easy*, *medium*, dan *hard*. Masing-masing tingkat akan membawa pengguna pada *map* permainan yang berbeda. Tampilan pemilihan tingkat ditunjukkan pada Gambar 4.2.



Gambar 4.2 Tampilan Pemilihan Tingkat

4.2.3 Implementasi Tampilan Bermain Permainan

Tampilan bermain permainan adalah tampilan paling kompleks yang terdapat dalam permainan *Bangjo Puzzle*. Pada tampilan ini sistem akan menampilkan objek-objek penyusun permainan yang diproses dengan menggunakan metode-metode yang telah dirancang. Tampilan bermain permainan adalah tampilan yang paling sering diakses karena pada tampilan inilah permainan *Bangjo Puzzle* dimainkan. Terdapat sebuah kontrol permainan yang digunakan selama proses bermain. Tombol *restart* tersedia diantara tombol kursor. Tampilan bermain permainan ditunjukkan pada Gambar 4.3.



Gambar 4.3 Tampilan Bermain Permainan

4.2.4 Implementasi Tampilan *Instruction*

Tampilan ini akan muncul jika pengguna memilih menu *Instruction* yang terdapat pada menu utama. Tampilan *instruction* berisi tentang cara bermain permainan *Bangjo Puzzle* secara singkat. Tidak ada

submenu lagi pada tampilan ini. Sehingga apabila pengguna sudah selesai dengan keperluannya, pengguna harus menekan ikon *close* yang terdapat pada sudut kanan atas layar. Tampilan *instruction* ditunjukkan pada Gambar 4.4.



Gambar 4.4 Tampilan *Instruction*

4.2.5 Implementasi Tampilan *Credit*

Tampilan *credit* adalah tampilan yang bertujuan untuk mengeluarkan pihak-pihak yang terlibat dalam pembuatan permainan *Bangjo Puzzle*. Tampilan *credit* diadakan sebagai tanda rasa terima kasih penulis kepada pihak-pihak yang telah membantu pembangunan permainan *Bangjo Puzzle*. Tampilan *credit* ditunjukkan pada Gambar 4.5.



Gambar 4.5 Tampilan *Credit*

4.2.6 Implementasi Kasus Penggunaan Memilih Tingkat Permainan

Terdapat tiga jenis tingkat yang tersedia di dalam permainan *Bangjo Puzzle*, yaitu *easy*, *medium*, dan *hard*. Masing-masing tingkat mempunyai ukuran *map* dengan matriks yang berbeda.. Jumlah matriks pada tingkat *easy*, *medium*, dan *hard* secara berturut-turut adalah delapan, sepuluh, dan dua belas yang selanjutnya disimpan pada variabel “GRID_COUNT” dan diolah pada fungsi selanjutnya. Kode Sumber 4.1 menunjukkan inisialisasi jumlah matriks pada tiga tingkat.

```

        global.TINGKATS =ceil( ini_read_real("Sound",
"tingkat", 0));
        global.SKORS =
        ini_read_real("Game"+string(global.TINGKATS) , "skor",
        0);
        global.GRID_COUNT =
        ini_read_real("Game"+string(global.TINGKATS) , "grids",
        10);
        global.TRY_COUNT =
        ini_read_real("Game"+string(global.TINGKATS) , "tries",
        5);

```

Kode Sumber 4.1 Inisialisasi Matriks Tingkat

4.2.7 Implementasi Kasus Penggunaan Mengulangi Tingkat Skenario

Mengulangi tingkat skenario adalah salah satu implementasi penggunaan yang cukup penting. Fungsi dari *restart* skenario ini adalah untuk menampilkan keadaan awal dari skenario terakhir yang sedang dijalankan. Matriks *map* akan disimpan dalam variabel sebagai *backup* dan akan ditampilkan ketika fungsi ini digunakan. Kode Sumber 4.2 digunakan untuk mengulangi tingkat skenario.

```

if (global.IS_RESTART){
    for(var j=0;j<global.GRID_COUNT ;j++){
        for(var i=0;i< global.GRID_COUNT ;i++){
            __map[# i,j] = global.ori_map[# i,j] ;
        }
    }
}

room_restart();
global.IS_RESTART = true;
scrGameTingkat();

```

Kode Sumber 4.2 Mengulangi Tingkat Skenario

4.2.8 Implementasi Kasus Penggunaan Melihat Instruksi Permainan

Fungsi ini digunakan untuk menampilkan menu *instruction*. Menu *instruction* dapat diakses melalui pada menu utama permainan dan memilih submenu *Instruction*. Kode Sumber 4.3 digunakan untuk menampilkan instruksi permainan.

```

/*--INSTRUCTION--*/

//Navigation into INSTRUCTION

room_goto(instruct);

```

Kode Sumber 4.3 Melihat Instruksi Permainan

4.2.9 Implementasi Kasus Penggunaan Melihat *Credit* Permainan

Fungsi ini digunakan untuk menampilkan menu *credit*. Menu *credit* dapat diakses melalui pada menu utama permainan dan memilih submenu *Credit*. Kode Sumber 4.4 digunakan untuk Menampilkan *credit* permainan.

```

/*--CREDIT--*/

//Navigation into CREDIT

room_goto(credit);

```

Kode Sumber 4.4 Melihat *Credit* Permainan

4.2.10 Implementasi *Midpoint Displacement*

Midpoint Displacement adalah sebuah metode yang digunakan sebagai pengacakan dengan cara mencari angka-angka pada semua posisi matriks. Implementasi *Midpoint Displacement* ditunjukkan pada Kode Sumber 4.5.

```
var x1 = argument0;
var y1 = argument1;
var x2 = argument2;
var y2 = argument3;

if(x2-x1<2 && y2-y1<2) return false;

// Find distance between points and use when generating
a random number.
var dist=(x2-x1+y2-y1);
// Find midpoint
var midx=(x1+x2) / 2;
var midy=(y1+y2) / 2;
// Get vals of corners
var c1=global.map[x1, y1];
var c2=global.map[x2, y1];
var c3=global.map[x2, y2];
var c4=global.map[x1, y2];

randomize();

if(global.map[midx,y1]==0)
global.map[midx,y1]=((c1+c2)/2) + random(dist);
if(global.map[midx,y2]==0)
global.map[midx,y2]=((c4+c3)/2) + random(dist);
if(global.map[x1,midy]==0)
global.map[x1,midy]=((c1+c4)/2) + random(dist);
if(global.map[x2,midy]==0)
global.map[x2,midy]=((c2+c3)/2) + random(dist);

global.map[midx,midy] = ((c1+c2+c3+c4+random(dist)) /
4);

midpoint(x1,y1,midx,midy);
midpoint(midx,y1,x2,midy);
midpoint(x1,midy,midx,y2);
midpoint(midx,midy,x2,y2);

return true;
```

Kode Sumber 4.5 *Midpoint Displacement*

4.2.11 Implementasi Penentuan Jalur

Penentuan jalur dilakukan setelah proses *Midpoint Displacement* selesai dilakukan. Hasil yang didapatkan pada saat proses *Midpoint Displacement* selanjutnya dilakukan modifikasi berupa pengubahan angka nol dan satu agar didapatkan jalan yang lebih banyak. Selanjutnya dilakukan penentuan posisi *block start* yang dilakukan secara acak pada matriks yang mempunyai nilai nol. Langkah selanjutnya adalah mengecek setiap kemungkinan jalur pada awal *start* dan membandingkan dengan kemungkinan jalur lainnya. Pengecekan kemungkinan jalur dilakukan hingga lima belas kali. Titik yang memiliki Jalur Penyelesaian paling banyak, selanjutnya dipilih sebagai jalur yang ditampilkan kepada pengguna. Apabila semua titik menghasilkan Jalur Penyelesaian kurang dari empat, maka sistem akan kembali melakukan proses *Midpoint Displacement*. Metode yang digunakan dalam proses ini adalah *Brute Force*. Kode Sumber 4.6 menunjukkan implementasi penentuan jalur.

```

        for(var j=0;j< global.GRID_COUNT ;j++){
            for(var i=0;i< global.GRID_COUNT ;i++){
                {
                    if      (      global.map[i,j]>3      &&
global.map[i,j]<5) {
                        __map [# i,j] = 1 ;
                    }else {
                        __map [# i,j] = 0 ;
                    }
                }
            }
        }
        setcoordinates();
randomize();
        var go_top = false,go_down = false,go_left =
        false,go_right = false;
        var isOK = false;
        var lastMove = 0;
        var mode_x = __sx;
        var mode_y = __sy;

        curr_y = __sy;
        curr_x = __sx;

```



```

    path=0;
    path1=0;
    path2=0;
    path3=0;
    var isTry = false;
    solCount = 0;
    var origin_y = __sy;
    var origin_x = __sx;
    //var rutes = ds_list_create();
    //VALIDATE RULE BY MOVE OBJECT
    brute_force_up();
    brute_force_left();
    brute_force_right();
    brute_force_dw();
    show_debug_message("CEK"+string(path)+"
"+string(path1)+" "+string(path2)+" "+string(path3))
    var pathmax=max(path,path1,path2,path3);

    if(pathmax=path){
        brute_force_up();
    }
    else if(pathmax=path1){
        brute_force_right();
    }
    else if(pathmax=path2){
        brute_force_dw();
    }
    else{
        brute_force_left();
    }
    //CHECKING SOLUTION
    show_debug_message("Solusi : " +string(pathmax));
    if (pathmax<=4){
        isSolved = false;
    }else {
        isSolved = true;
    }
    if (origin_x == curr_x or origin_y==curr_y){
        isSolved = false;
    }
}

```

Kode Sumber 4.6 Implementasi Penentuan Jalur

(Halaman Ini Sengaja Dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Bab ini akan menjelaskan rangkaian pengujian permainan *Bangjo Puzzle* berikut dengan evaluasinya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba perangkat lunak meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pengujian dicantumkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak[6]

Perangkat Keras	<i>Device</i> : Sony Xperia M2 Aqua RAM : 1 Gb Internal Memori : 8 Gb Resolusi Layar : 540 x 960 pixel
Perangkat Lunak	Sistem Operasi : Google Android 5.1 (Lollipop)

5.2 Pengujian Pembangkitan *Puzzle*.

Pengujian ini bertujuan untuk mengetahui tingkat kedinamisan *map* yang dihasilkan oleh komputer. Pengujian dilakukan pada halaman Bermain Permainan seperti yang telah dijelaskan pada Subbab 3.2.3.3. Pada halaman ini *map* akan dilakukan *generate* secara acak.

5.2.1 Skenario Uji Pembangkitan *Puzzle*

Pengujian ini dilakukan untuk mengetahui tingkat keberagaman *map* yang dihasilkan. Skenario pengujian pembangkitan *puzzle* disajikan dalam Tabel 5.2.

Tabel 5.2 Skenario Pengujian Pembangkitan *Puzzle*

Deskripsi	Bertujuan untuk mengetahui tingkat keberagaman <i>map</i> yang dihasilkan oleh sistem.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>Play Game</i> pada menu utama permainan. 2. Pengguna melakukan pemilihan tingkat. 3. Pengguna melakukan pencatatan data <i>map</i> yang telah dihasilkan. 4. Pengguna melakukan langkah kedua dan ketiga secara berulang hingga dua puluh kali. 5. Hasil dari pengujian dituliskan dalam Tabel 5.3.

Pengujian dilakukan pada semua tingkat untuk mengetahui keberagaman *map* yang telah dihasilkan. Objek keberagaman *map* yang diuji yaitu, pola *obstacle*, *block start*, *block finish*, dan bintang. Hasil pengujian akan dibahas pada Subbab 5.2.2.

5.2.2 Hasil Pengujian Pembangkitan *Puzzle*

Hasil pengujian pembangkitan mencatat setiap pola *obstacle*, *block start*, *block finish*, dan bintang yang memiliki kemiripan identik. Percobaan dilakukan sebanyak dua puluh kali pada masing-masing tingkat. Hasil uji coba ditunjukkan pada Tabel 5.3.

Tabel 5.3 Hasil Uji Coba Pembangkitan *Puzzle*

No	Tingkat	Frekuensi <i>Puzzle</i> dengan Pola Sama
1	<i>Easy</i>	Tidak Pernah
2	<i>Medium</i>	Tidak Pernah
3	<i>Hard</i>	Tidak Pernah

Tabel 5.3 menunjukkan bahwa kemunculan *map* dengan kemiripan identik memiliki frekuensi nol. Hal ini menunjukkan bahwa implementasi pengacakan *map* menggunakan metode *Midpoint Displacement* berhasil dilakukan. Tangkapan layar hasil pengujian selanjutnya disajikan dalam Lampiran 1.

5.3 Pengujian Aturan Permainan *Bangjo Puzzle*

Pengujian aturan permainan bertujuan untuk mengetahui apakah permainan telah menerapkan komponen-komponen sesuai dengan perancangan yang terdapat dalam Subbab 3.2.5.

5.3.1 Skenario Uji Aturan Permainan *Bangjo Puzzle*

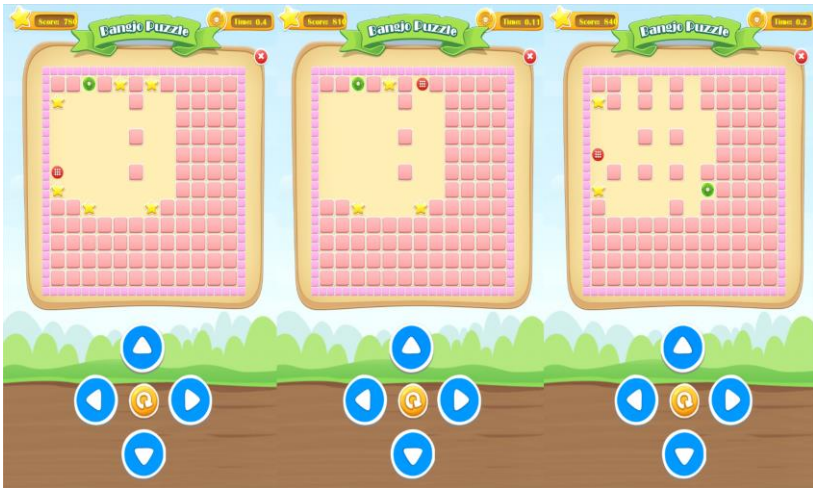
Skenario pengujian digunakan untuk menguji aturan permainan *Bangjo Puzzle*. Skenario pengujian aturan permainan disajikan dalam Tabel 5.4.

Tabel 5.4 Skenario Pengujian Aturan Main

Deskripsi	Bertujuan untuk mengetahui performa sistem dalam menerapkan aturan permainan.
Skenario Pengujian	<ol style="list-style-type: none"> 1. Pengguna memindahkan <i>block start</i>. 2. Sistem menggerakkan <i>block start</i> secara lurus dan berhenti ketika menabrak <i>obstacle</i> atau berada di ujung papan. 3. Pengguna mengambil bintang. 4. Sistem memberikan tanda keberhasilan pemain mengambil bintang. 5. Pengguna melakukan kesalahan gerakan yang mengakibatkan gerakan buntu sehingga diharuskan melakukan <i>restart</i>. 6. Sistem melakukan <i>restart</i> pada skenario.

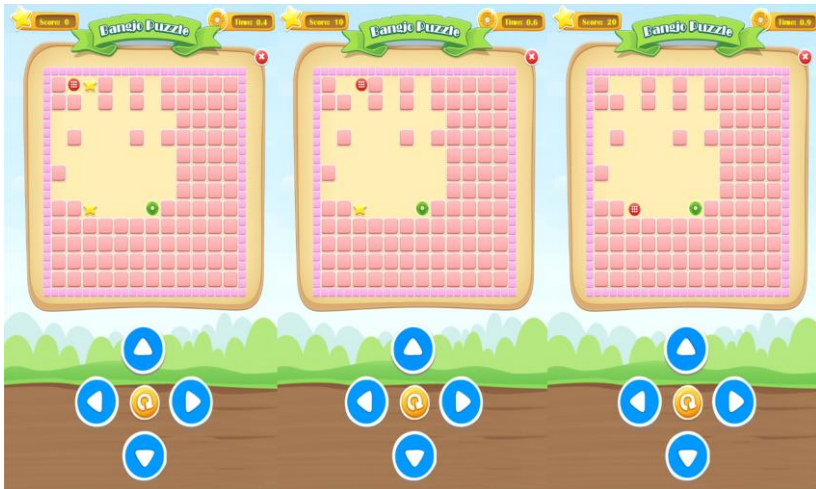
5.3.2 Hasil Pengujian Aturan Permainan *Bangjo Puzzle*

Hasil pengujian aturan permainan *Bangjo Puzzle* disajikan dalam gambar-gambar di bawah ini. Pengujian yang dilakukan mengacu pada skenario pengujian aturan main yang telah dijelaskan pada Tabel 5.4. Gambar 5.1 menunjukkan bahwa sistem menggerakkan *block start* secara lurus dan berhenti ketika menabrak *obstacle* atau berada di ujung papan.



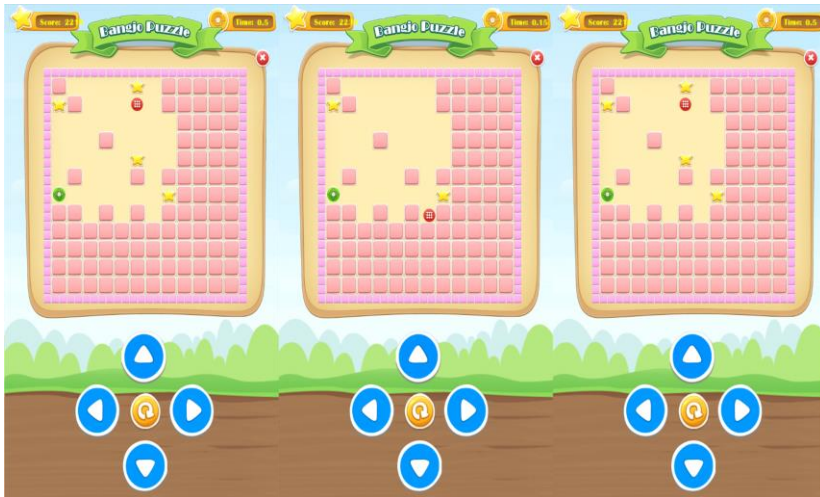
Gambar 5.1 Sistem Menggerakkan *Block Start*

Skenario pengujian selanjutnya adalah pengguna mengambil bintang yang disediakan oleh sistem. Sistem memberikan sebuah tanda berupa *sound alert* apabila pemain berhasil mengambil bintang. Skenario tidak akan terselesaikan apabila pemain masih menyisakan bintang di papan permainan. Sistem akan menampilkan skenario berikutnya apabila semua bintang telah diambil dan pemain telah berhasil memindahkan *block start* menuju *block finish*. Gambar 5.2 menunjukkan skenario pengujian pengambilan bintang oleh pemain.



Gambar 5.2 Pemain Mengambil Bintang

Pada Gambar 5.3 pemain melakukan kesalahan gerakan yang berakibat pada terkuncinya *block start*. Pemain kemudian melakukan *restart* skenario dan mengulangi skenario dari awal.



Gambar 5.3 Sistem Melakukan *Restart* Skenario

5.4 **Pengujian Penyelesaian Skenario**

Pengujian ini bertujuan untuk mengetahui tingkat skenario yang gagal terselesaikan. Pengujian ini juga merupakan pengujian metode Algoritma *Brute Force* dalam menentukan jalur penyelesaian.

5.4.1 **Skenario Uji Penyelesaian Skenario**

Skenario pengujian dilakukan dengan cara meyelesaikan skenario pada setiap tingkat sebanyak sepuluh perulangan. Detail penyelesaian tiap skenario selanjutnya akan dibahas pada Subbab 5.4.2.

5.4.2 **Hasil Uji Penyelesaian Skenario**

Hasil pengujian mencatat setiap detail skenario yang meliputi, jumlah langkah, jumlah bintang, dan jumlah *obstacle*. Tabel 5.5 menunjukkan hasil pengujian penyelesaian skenario.

Tabel 5.5 Hasil Pengujian Penyelesaian Skenario

Tingkat	Percobaan Ke-	Dapat Diselesaikan	Jumlah		
			Langkah	<i>Obstacle</i>	Bintang
<i>Easy</i>	1	YA	4	13	3
	2	YA	7	12	5
	3	YA	9	12	3
	4	YA	10	18	5
	5	YA	11	10	7
	6	YA	4	17	2
	7	YA	10	16	7
	8	YA	7	11	4
	9	YA	11	14	6
	10	YA	10	15	6
<i>Medium</i>	1	YA	10	40	6
	2	YA	8	39	3
	3	YA	11	31	5

	4	YA	10	36	5
	5	YA	13	29	7
	6	YA	11	43	7
	7	YA	8	29	6
	8	YA	4	41	4
	9	YA	5	41	3
	10	YA	10	37	6
<i>Hard</i>	1	YA	9	34	5
	2	YA	10	26	5
	3	YA	16	21	9
	4	YA	10	30	6
	5	YA	9	28	5
	6	YA	11	39	6
	7	YA	7	37	3
	8	YA	9	18	5
	9	YA	14	35	8
	10	YA	12	29	7

5.5 Pengujian Fungsionalitas Permainan

Pengujian ini bertujuan untuk mengetahui kesesuaian fungsionalitas aplikasi dengan perancangan yang telah dilakukan pada Bab 3. Pengujian yang akan dilakukan adalah dengan menggunakan metode *Black Box*, yaitu sebuah metode pengujian yang dilakukan dengan berfokus pada fungsionalitas aplikasi[7].

5.5.1 Skenario Uji Fungsionalitas Permainan

Skenario pengujian fungsionalitas permainan bertujuan untuk mengecek kesesuaian perancangan dan hasil yang diharapkan. Skenario pengujian fungsionalitas ditunjukkan pada Tabel 5.6.

Tabel 5.6 Skenario Pengujian Fungsionalitas

Tahap Pengujian	Deskripsi
Kondisi Awal	Pengguna menjalankan sistem dan masuk pada halaman utama.
Prosedur Pengujian	Pengguna melakukan pengujian sesuai dengan perancangan fungsionalitas pada Subbab 3.2.1.
Hasil yang Diharapkan	Pengguna berhasil melakukan semua fungsionalitas sesuai dengan perancangan fungsionalitas pada Subbab 3.2.1.

5.5.2 Pengujian Fungsionalitas Kasus Penggunaan

Terdapat lima pengujian fungsionalitas kasus penggunaan yang dilakukan pada permainan *Bangjo Puzzle*, yaitu memilih jenis tingkat, menyelesaikan tingkat skenario, mengulangi tingkat skenario, melihat instruksi permainan, dan melihat *credit* permainan.

5.5.2.1 Pengujian Kasus Penggunaan Memilih Tingkat Permainan

Pengujian dilakukan dengan cara memilih jenis tingkat pada awal permainan. Terdapat tiga macam tingkat yang dapat dipilih, yaitu *easy*, *medium*, dan *hard*. Tingkat *easy*, *medium*, dan *hard* secara berturut-turut akan menampilkan papan dengan matriks persegi delapan, sepuluh, dan dua belas. Pengujian dikatakan berhasil apabila sistem dapat menampilkan papan sesuai dengan perancangan. Gambar 5.4 adalah tampilan pengujian kasus penggunaan memilih tingkat permainan.



Gambar 5.4 Tampilan Pengujian Kasus Penggunaan Memilih Tingkat Permainan

5.5.2.2 Pengujian Kasus Penggunaan Bermain Permainan

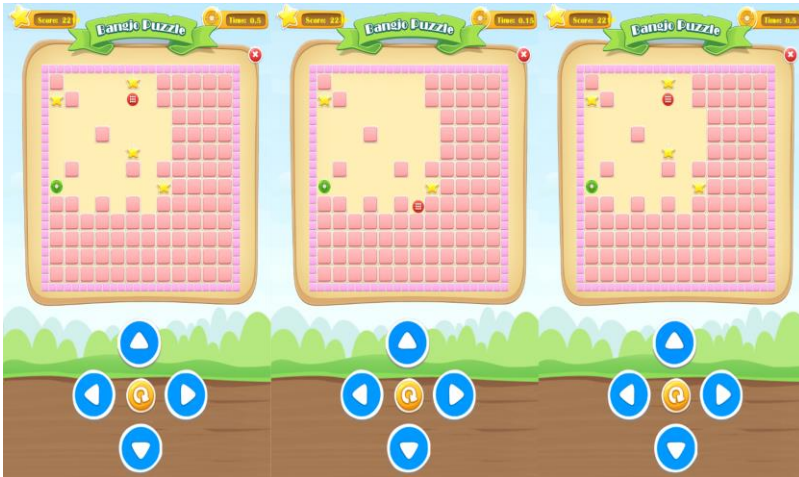
Pengujian dilakukan dengan cara bermain permainan. Pengujian dianggap berhasil apabila skenario telah menerapkan semua aturan permainan yang telah dirancang. Gambar 5.5 adalah tampilan pengujian kasus penggunaan bermain permainan.



Gambar 5.5 Tampilan Pengujian Kasus Penggunaan Bermain Permainan

5.5.2.3 Pengujian Kasus Penggunaan Mengulangi Tingkat Skenario

Pengujian dilakukan dengan cara mengulang skenario yang sedang dijalankan. Pengujian dianggap berhasil apabila sistem dapat mengulangi skenario dan pemain kembali melanjutkan permainan. Gambar 5.6 adalah tampilan pengujian kasus penggunaan mengulangi tingkat skenario.



Gambar 5.6 Tampilan Pengujian Kasus Penggunaan Mengulangi Tingkat Skenario

5.5.2.4 Pengujian Kasus Penggunaan Melihat Instruksi Permainan

Pengujian dilakukan dengan cara memilih menu *Instruction* pada menu utama permainan. Pengujian dianggap berhasil apabila sistem menampilkan halaman *instruction*. Gambar 5.7 adalah tampilan pengujian melihat instruksi permainan.



Gambar 5.7 Tampilan Pengujian Kasus Penggunaan Melihat Instruksi Permainan

5.5.2.5 Pengujian Kasus Penggunaan Melihat *Credit* Permainan

Pengujian dilakukan dengan cara memilih menu *Credit* pada menu utama permainan. Pengujian dianggap berhasil apabila sistem menampilkan halaman *Credit*. Gambar 5.8 adalah tampilan pengujian kasus penggunaan melihat *credit* permainan.



Gambar 5.8 Tampilan Pengujian Kasus Penggunaan Melihat *Credit* Permainan

5.5.3 Hasil Pengujian Fungsionalitas

Hasil pengujian fungsionalitas menunjukkan bahwa semua perancangan kasus penggunaan yang terdapat dalam Subbab 3.2.1 telah terimplementasi dengan baik pada permainan. Hasil pengujian fungsionalitas selanjutnya dijabarkan dalam Tabel 5.7.

Tabel 5.7 Hasil Pengujian Fungsionalitas

No	Kode Kasus Penggunaan	Hasil Pengujian
1	UC-001	Berhasil
2	UC-002	Berhasil
3	UC-003	Berhasil
4	UC-004	Berhasil
5	UC-005	Berhasil

5.6 Pengujian Pengguna

Pengujian pengguna merupakan rangkaian pengujian terakhir permainan *Bangjo Puzzle*. Tujuan diadakannya pengujian pengguna adalah sebagai media penghimpun masukan-masukan bersifat subyektif yang berasal dari berbagai macam individu dengan latar belakang berbeda. Masukan-masukan ini sangat penting untuk didapatkan agar permainan *Bangjo Puzzle* dapat dinikmati oleh semua kalangan.

5.6.1 Skenario Uji Pengguna

Skenario yang digunakan dalam pengujian pengguna adalah dengan meminta pengguna memainkan aplikasi sesuai dengan kemampuannya. Pengguna berasal dari berbagai macam latar belakang, sehingga diharapkan didapatkan data sesubjektif mungkin.

Pengguna akan diberikan edukasi mengenai gambaran umum permainan. Jumlah pengguna yang terlibat adalah sebanyak sepuluh orang. Selanjutnya pengguna diarahkan untuk mengisi kuisioner yang berkaitan tentang penilaian permainan. Hasil dari

penilaian kuisioner sangat diperlukan untuk perbaikan fitur dan perkembangan permainan ke depannya.

Penilaian permainan terdiri dari tiga parameter, yaitu antarmuka, kedinamisan permainan, dan kenyamanan. Masing-masing parameter memiliki beberapa pertanyaan yang harus dinilai berdasarkan empat kategori, yaitu kategori sangat tidak setuju, tidak setuju, setuju, dan sangat setuju. Keempat kategori ini kemudian disubstitusi dengan angka satu untuk penilaian terendah dan empat untuk penilaian tertinggi. Masing-masing penilaian nantinya akan dikalikan dua koma lima agar didapatkan nilai angka maksimal seratus.

5.6.2 Daftar Penguji Perangkat Lunak

Pengguna yang dalam hal ini adalah penguji perangkat lunak, terdiri dari berbagai macam latar belakang. Hal ini bertujuan agar didapatkan penilaian sesubjektif mungkin yang sangat diperlukan dalam pengembangan permainan selanjutnya. Tabel 5.8 menunjukkan daftar nama penguji perangkat lunak.

Tabel 5.8 Daftar Nama Penguji Perangkat Lunak

No	Nama	Pekerjaan
1	Agung Teguh Setyadi	Mahasiswa
2	Ahmad Ridwan Fauzi	Mahasiswa
3	Ivan Adrian Imantaka	Penulis
4	Dimas Putri Supawesti	Mahasiswa
5	Reva Girindra Ariyadi	Seniman
6	M. Yusron Irsyadi	Karyawan
7	Burhan Ramadhan	Pengangguran
8	Alfin Nur Widikdo	Wiraswasta
9	Yusuf Aziz Henny T.Y	Karyawan
10	Rifki Maula Iqbal	Wiraswasta

5.6.3 Hasil Penilaian Pengguna

Penilaian permainan terdiri dari tiga parameter, yaitu parameter antarmuka, kedinamisan permainan, dan kenyamanan. Masing-

masing parameter memiliki beberapa pernyataan yang harus dinilai. Penilaian pernyataan yang terdapat pada parameter berdasarkan empat kategori, yaitu sangat tidak setuju, tidak setuju, setuju, dan sangat setuju. Berikut dijabarkan hasil penilaian permainan yang dilakukan dengan media kuisioner pada sepuluh responden.

5.6.3.1 Hasil Penilaian Antarmuka

Penilaian antarmuka adalah penilaian yang berkaitan dengan tampilan permainan secara keseluruhan. Pengguna diharuskan memberikan penilaian pada desain tampilan permainan secara keseluruhan. Hasil penilaian antarmuka ditunjukkan dalam Tabel 5.9.

Tabel 5.9 Penilaian Antarmuka

No.	Antarmuka	Penilaian (P)				Nilai (Px2,5)
		1	2	3	4	
1	Aplikasi memiliki tata letak tombol menu yang mudah dilihat/dikenali.	0	1	3	6	87,5
2	Aplikasi memiliki tampilan, warna, dan desain yang menarik.	0	1	5	4	82,5
Nilai Akhir (Jumlah Nilai / Jumlah Pernyataan)						85

5.6.3.2 Hasil Penilaian Kedinamisan Permainan

Penilaian kedinamisan permainan adalah penilaian yang dilakukan untuk mengetahui dampak yang dirasakan pengguna yang dihasilkan dari kedinamisan permainan. Terdapat enam pernyataan yang digunakan dalam penilaian ini. Hasil penilaian kedinamisan permainan ditunjukkan pada Tabel 5.10.

Tabel 5.10 Penilaian Kedinamisan Permainan

No.	Kedinamisan Permainan	Penilaian (P)				Nilai (Px2,5)
		1	2	3	4	
1	Saya baru kali ini memainkan Permainan <i>Puzzle</i> dengan skenario yang dinamis.	1	1	3	5	80
2	Saya merasa tertantang dengan skenario Permainan <i>Puzzle Dinamis</i> .	0	1	4	5	85
3	Saya merasa tidak cepat bosan dalam bermain Permainan <i>Puzzle Dinamis</i> .	0	1	3	6	87,5
4	Skenario Permainan <i>Puzzle</i> membantu saya dalam mengisi kekosongan waktu saya.	0	1	5	4	82,5
5	Saya merasa terbantu dalam mengasah pikiran saat menyelesaikan skenario yang dinamis.	0	2	5	3	77,5
6	Saya merasa lebih mengetahui proses dari Permainan <i>Puzzle Dinamis</i> .	0	2	4	4	80
Nilai Akhir (Jumlah Nilai / Jumlah Pernyataan)						82

5.6.3.3 Hasil Penilaian Kenyamanan

Penilaian kenyamanan adalah penilaian yang dilakukan terhadap faktor eksternal selama pengguna menjalankan aplikasi. Terdapat dua pernyataan yang digunakan dalam penilaian ini. Hasil penilaian kenyamanan ditunjukkan pada Tabel 5.11.

Tabel 5.11 Penilaian Kenyamanan

No.	Kenyamanan	Penilaian (P)				Nilai (Px2,5)
		1	2	3	4	
1	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan atau <i>hang</i> .	0	0	2	8	95
2	Saya merasa terbantu dengan adanya petunjuk yang disediakan.	0	2	5	3	77,5
Nilai Akhir (Jumlah Nilai / Jumlah Pernyataan)						86,25

5.6.4 Kesimpulan Hasil Pengujian Pengguna

Hasil penilaian yang dilakukan pada permainan *Bangjo Puzzle* menghasilkan nilai minimal delapan puluh dua dan maksimal 86,25. Hal ini menunjukkan bahwa sebagian besar pengguna telah memberikan apresiasi yang baik terhadap kehadiran permainan *Bangjo Puzzle*. Apresiasi yang diberikan oleh pengguna mencakup aspek antarmuka, kedinamisan permainan, dan tingkat kenyamanan. Dengan demikian, tujuan dari penyusunan tugas akhir ini dapat dikatakan tercapai melalui parameter-parameter di atas.

BAB VI

KESIMPULAN DAN SARAN

Bab ini adalah bab terakhir yang akan dibahas dalam penyusunan tugas akhir ini. Pada bab ini akan dibahas mengenai kesimpulan yang didapatkan dari hasil perancangan, implementasi, beserta pengujian permainan *Bangjo Puzzle*. Selain itu, ditambahkan juga beberapa saran yang digunakan untuk pengembangan permainan ini selanjutnya.

6.1. Kesimpulan

Proses penyusunan tugas akhir yang terdiri dari analisis, perancangan, implementasi, dan pengujian menghasilkan beberapa kesimpulan yang akan dijabarkan pada poin-poin di bawah ini:

1. Pengujian aturan permainan menunjukkan bahwa sistem telah berhasil menerapkan aturan permainan yang telah dirancang. Hal ini dibuktikan dengan semua aturan main telah terdapat di dalam permainan *Bangjo Puzzle*.
2. Pengujian pembangkitan *puzzle* menunjukkan bahwa sistem telah berhasil melakukan *generate map* secara dinamis menggunakan metode *Midpoint Displacement*. Hal ini dibuktikan dengan tidak adanya frekuensi pola yang sama dengan percobaan selama dua puluh kali.
3. Pengujian penyelesaian skenario menunjukkan bahwa setiap skenario yang dihasilkan oleh sistem dapat terselesaikan dengan sukses menggunakan metode algoritma *Brute Force*. Hal ini dibuktikan dengan percobaan selama sepuluh kali pada masing-masing tingkat dan menghasilkan skenario yang terselesaikan.

6.2. Saran

Saran yang diberikan untuk pengembangan permainan *Bangjo Puzzle* selanjutnya, dijabarkan dalam beberapa poin berikut ini:

1. Penggunaan *sound alert* harus lebih diperhitungkan lagi kemeriahannya. *Sound alert* yang terlalu meriah menyebabkan rasa kurang nyaman pada beberapa pengguna.
2. Penggunaan kontrol *swipe* untuk memindahkan blok akan lebih memudahkan pengguna.
3. Penggunaan *hint* perlu dipertimbangkan untuk memudahkan pengguna yang kesulitan memecahkan *puzzle*.
4. Penambahan jumlah jalur perlu dipertimbangkan sebagai acuan langkah bagi pengguna.

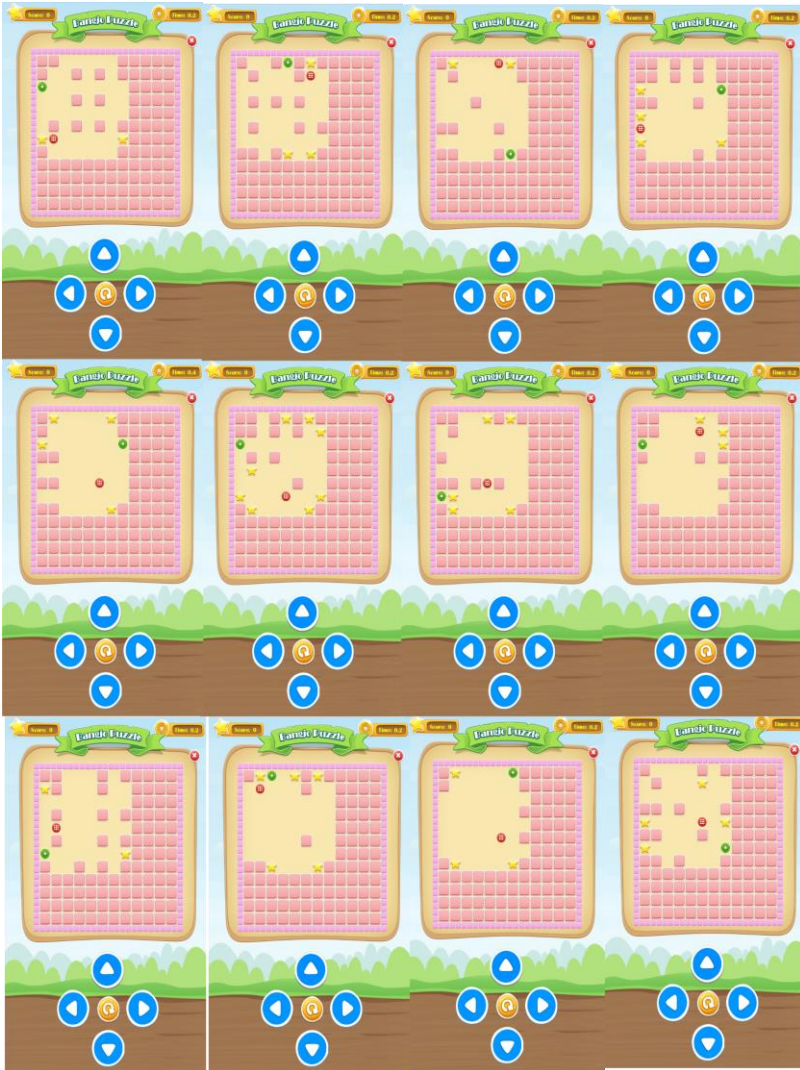
DAFTAR PUSTAKA

- [1] “About YoYo Games,” *Yoyo Games*. [Online]. Available: <http://www.yoyogames.com/about>. [Accessed: 01-Dec-2017].
- [2] “Features,” *Yoyo Games*. [Online]. Available: <http://www.yoyogames.com/gamemaker/features>. [Accessed: 01-Dec-2017].
- [3] “YoYo Account.” [Online]. Available: <https://account.yoyogames.com/downloads>. [Accessed: 01-Dec-2017].
- [4] Y. A. Pandunegoro, “Pembangkit World dan Tileset Dinamis pada Game BATTLE Of THRONE dengan Genre Turn Based Strategy Game Menggunakan Metode Midpoin,” Jan-2016. [Online]. Available: <http://webcache.googleusercontent.com/search?q=cache:7mDxM8dp3DwJ:repository.its.ac.id/48911/1/5111100182-Undergraduate%2520Thesis.pdf+%&cd=2&hl=id&ct=clnk&gl=id&client=firefox-b>. [Accessed: 01-Dec-2017].
- [5] B. W. Santoso, F. Sundawa, and M. Azhari, “Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database,” *Magister Ilmu Komput. Univ. Budi Luhur*, vol. 6 No. 1, Mar. 2016.
- [6] “Sony Xperia M2 Aqua - Full phone specifications.” [Online]. Available: https://www.gsmarena.com/sony_xperia_m2_aqua-6582.php. [Accessed: 16-Jan-2018].
- [7] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” vol. Teknik Informatika Universitas Widyatama, Agustus 2015.

(Halaman Ini Sengaja Dikosongkan)

LAMPIRAN 1

1. Pengujian Pembangkitan *Puzzle* Pada Tingkat *Easy*



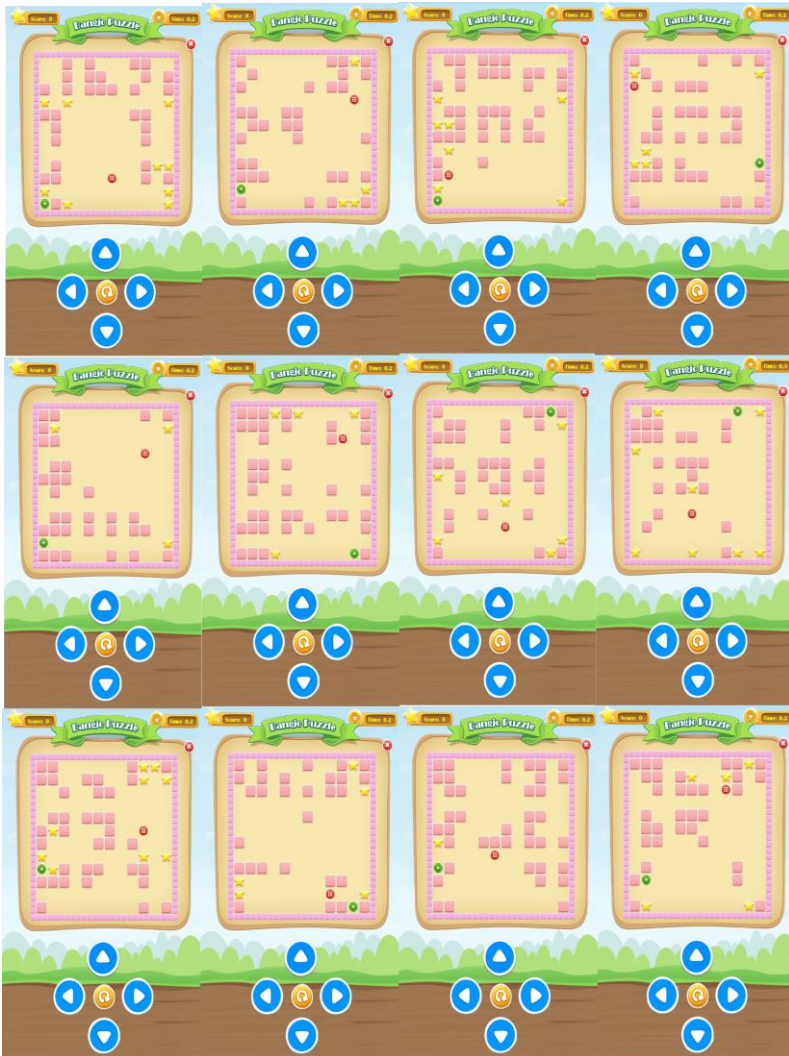


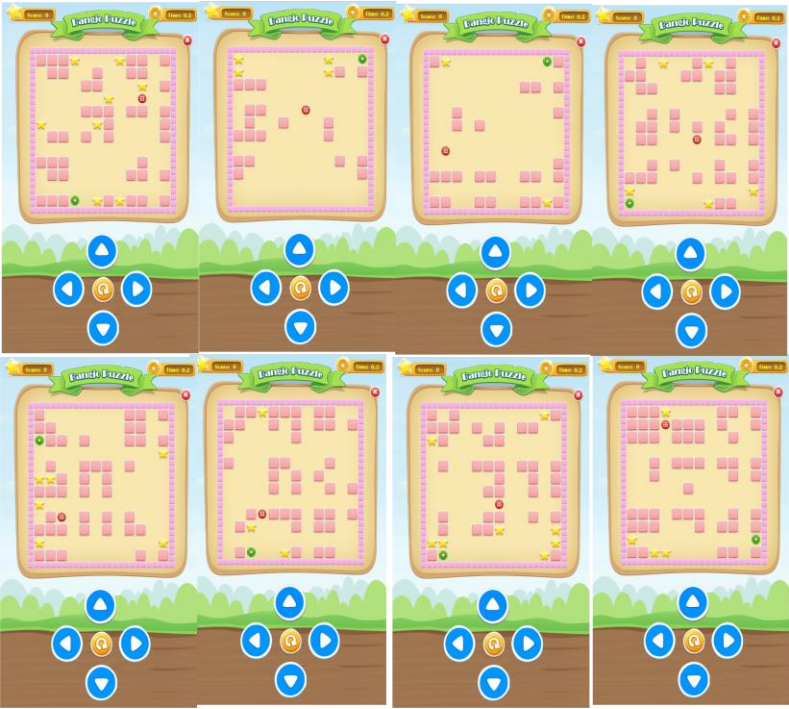
2. Pengujian Pembangkitan *Puzzle* Pada Tingkat *Medium*





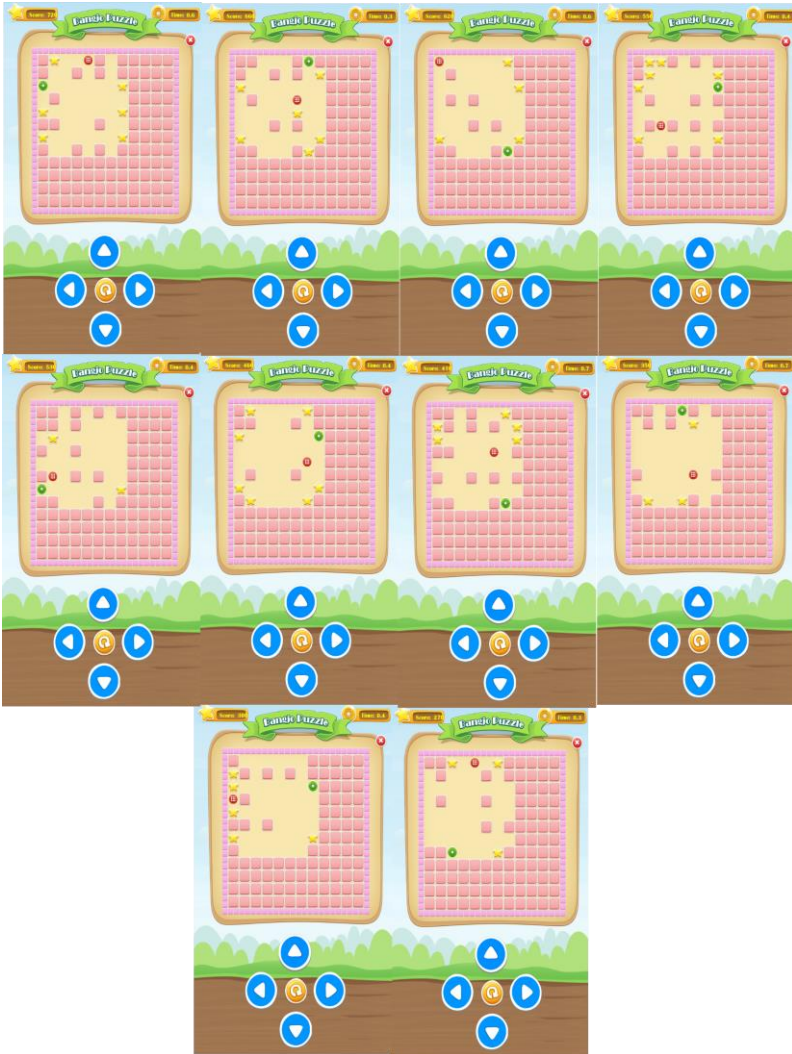
3. Pengujian Pembangkitan *Puzzle* Pada Tingkat *Hard*



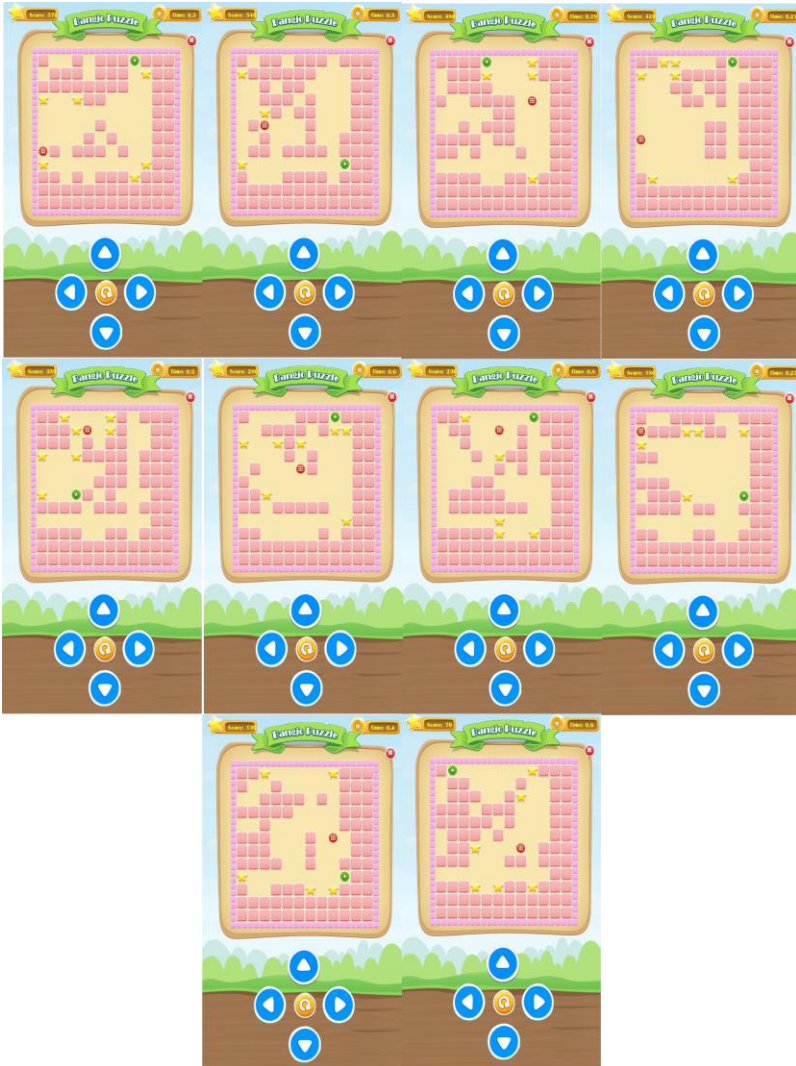


LAMPIRAN 2

1. Pengujian Penyelesaian Skenario Pada Tingkat *Easy*



2. Pengujian Penyelesaian Skenario Pada Tingkat *Medium*



3. Pengujian Penyelesaian Skenario Pada Tingkat *Hard*



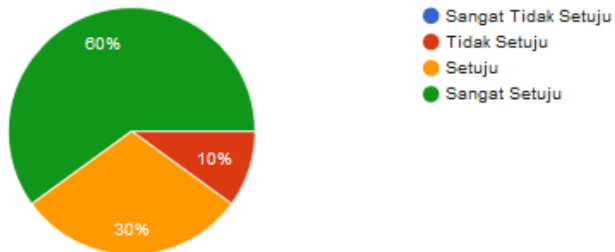
(Halaman Ini Sengaja Dikosongkan)

LAMPIRAN 3

1. Penilaian Antarmuka

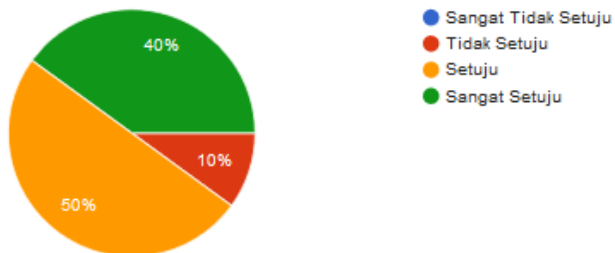
Aplikasi memiliki tata letak tombol menu yang mudah dilihat/dikenali.

10 responses



Aplikasi memiliki tampilan, warna, dan desain yang menarik.

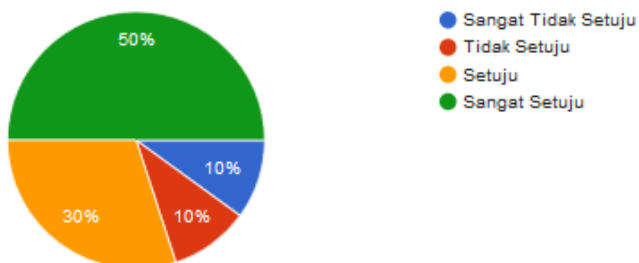
10 responses



2. Penilaian Kedinamisan Permainan

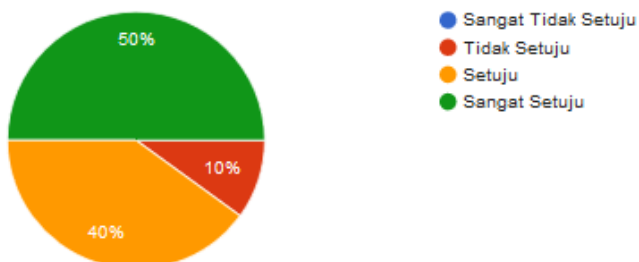
Saya baru kali ini memainkan Game Puzzle dengan skenario yang dinamis.

10 responses



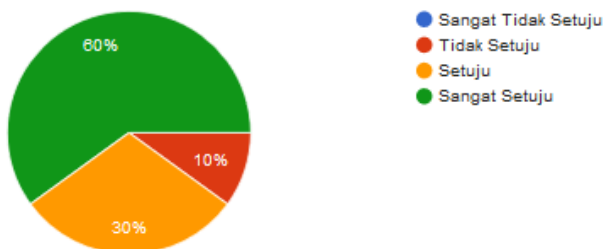
Saya merasa tertantang dengan skenario game Puzzle Dinamis.

10 responses



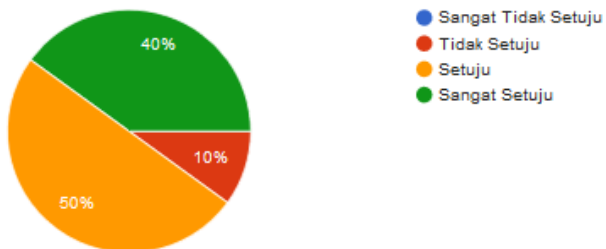
Saya merasa tidak cepat bosan dalam bermain game Puzzle Dinamis.

10 responses



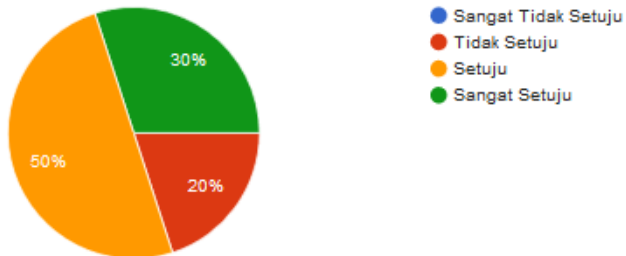
Skenario game Puzzle membantu saya dalam mengisi kekosongan waktu saya.

10 responses



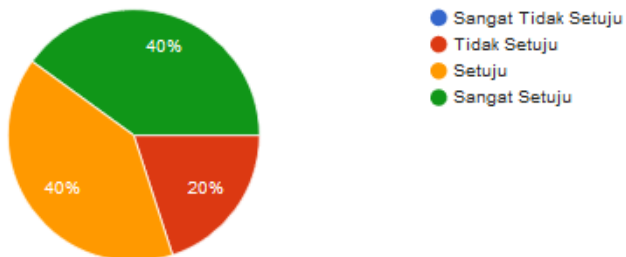
Saya merasa terbantu dalam mengasah pikiran saat menyelesaikan skenario yang dinamis.

10 responses



Saya merasa lebih mengetahui proses dari game Puzzle Dinamis.

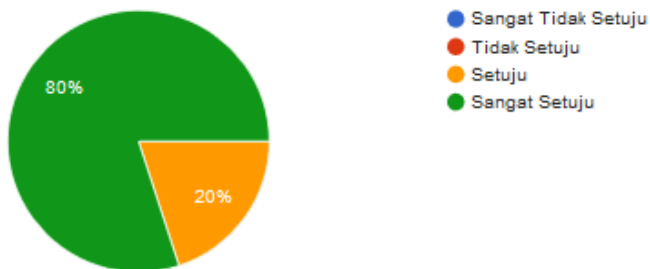
10 responses



3. Penilaian Kenyamanan

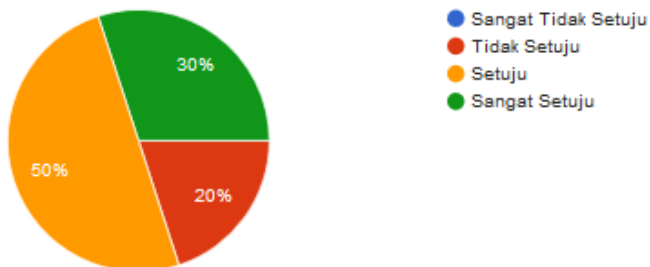
Aplikasi dapat berjalan lancar tanpa adanya lag dan atau hang.

10 responses



Saya merasa terbantu dengan adanya petunjuk yang disediakan.

10 responses



(Halaman Ini Sengaja Dikosongkan)

BIODATA PENULIS



Bulan Rahmat Setyadi atau biasa dipanggil Bulan dilahirkan di Jombang pada tanggal 08 Juni 1994. Penulis adalah anak pertama dari dua bersaudara. Walaupun mempunyai nama Bulan tetapi penulis adalah seorang laki-laki.

Penulis menempuh pendidikan dasar di SDN Jombatan V Jombang, pendidikan menengah pertama di SMP Negeri 2 Jombang, dan pendidikan menengah atas di SMA Negeri 2 Jombang. Setelah menyelesaikan studi di SMA, penulis melanjutkan pendidikan tinggi di Departemen Teknik Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Kegiatan non-akademik penulis selama duduk di bangku sekolah maupun kuliah adalah dengan menjadi anggota OSIS dengan jabatan Ketua Bidang Ketaqwaan Terhadap Tuhan Yang Maha Esa di SMP Negeri 2 Jombang dan staff Himpunan Mahasiswa Teknik Computer Informatika (ITS) pada departemen KMB. Selain itu, penulis juga aktif dalam mengikuti festival kreatif tentang perfilman dan periklanan dan sudah mendapat berbagai macam penghargaan, diantaranya Juara Umum *V-Kool Ceative Video Challenge* 2017, Silver Caraka Festival Keatif Mahasiswa 2014, Finalis Ajisaka Universitas Gadjah Mada Yogyakarta 2015, dan masih banyak yang lainnya.

Penulis tertarik pada bidang kreatif seperti desain, film, dan *game*. Selain itu, penulis juga mempunyai kegemaran dalam bidang kuliner, otomotif, dan fitness. Penulis dapat dihubungi pada alamat *e-mail* bulanrahmatsetyadi@gmail.com.